PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-342280

(43)Date of publication of application : 29.11.2002

(51)Int.Cl. G06F 15/00
G06F 9/46
G06F 13/00
G06F 15/16

(21)Application number : 2002-057064

(22)Date of filing : 04.03.2002

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>

(72)Inventor : BASKEY MICHAEL E
DEGILIO FRANK J
JONES JOHN C
ROHRBACH CHRISTIAN F
TEMPLE JOSEPH L III

(30)Priority

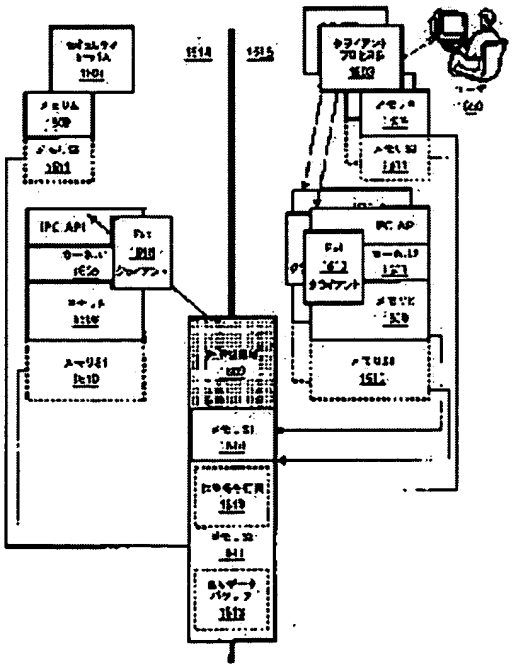
Priority number : 2001 801492 Priority date : 08.03.2001 Priority country : US

(54) PARTITIONED PROCESSING SYSTEM, METHOD FOR SETTING SECURITY IN THE SAME SYSTEM AND COMPUTER PROGRAM THEREOF

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an inter-partition message passing method for a security server in a partitioned processing environment.

SOLUTION: A common security server 1601 is run in a first partition 1614, and at least one security client 1603 is run in at least one other partition 1615, and each partition is provided with a shared memory or inter-memory connection to the first partition, so that a security client server can communicate with the common security server, and a mechanism connected to the security client for transmitting an authorization request by a user to the security client is included. The security



client transmits the authorization request via a main storage area 1609 to the common security server, and the common security server transmits a reply to the authorization request via the main storage area to the security client. Then, the security client transmits the reply to a user 1650.

LEGAL STATUS

[Date of request for examination]	04.03.2002
[Date of sending the examiner's decision of rejection]	07.09.2005
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	
[Date of registration]	
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号
特開2002－342280
（Ｐ２００２－３４２２８０Ａ）
(43)公開日　平成14年11月29日(2002.11.29)

(51)Int.Cl.[7]	識別記号	Ｆ Ｉ			テーマコード（参考）
G06F 15/00	330	G06F 15/00	330	A	5B045
9/46	350	9/46	350		5B085
13/00	351	13/00	351	Z	5B089
15/16	620	15/16	620	B	5B098

審査請求　有　請求項の数26　ＯＬ　（全22頁）

(21)出願番号　特願2002－57064(Ｐ2002－57064)

(22)出願日　平成14年３月４日(2002.3.4)

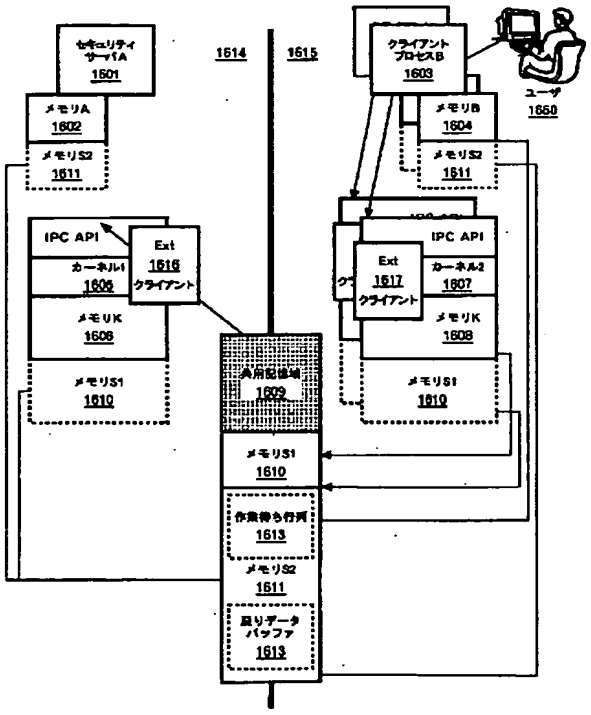
(31)優先権主張番号　０９／８０１４９２
(32)優先日　平成13年３月８日(2001.3.8)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国10504、ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100086243
弁理士　坂口　博　（外１名）

最終頁に続く

(54)【発明の名称】区分処理システム、区分処理システムにおけるセキュリティを設ける方法、およびそのコンピュータ・プログラム

(57)【要約】
【課題】　区分処理環境におけるセキュリティ・サーバのための区分間メッセージ受渡し方法を得る。
【解決手段】　共通セキュリティ・サーバ１６０１が第１の区分１６１４で実行され、少なくとも１つのセキュリティ・クライアント１６０３が、少なくとも１つの他の区分１６１５で実行され、各区分が共用メモリまたは第１の区分へのメモリ間接続を有し、それによってセキュリティ・クライアントが共通セキュリティ・サーバと通信することができ、セキュリティ・クライアントに接続され、ユーザによる認証要求をセキュリティ・クライアントに送る機構も含む。セキュリティ・クライアントが、認証要求を主記憶域１６０９を介して共通セキュリティ・サーバに送る。共通セキュリティ・サーバが、認証要求に対する応答を主記憶域を介してセキュリティ・クライアントに送る。次に、セキュリティ・クライアントがその応答をユーザ１６５０に送る。



1

【特許請求の範囲】

【請求項１】共通セキュリティ・サーバを含む第１の区分とセキュリティ・クライアントを含む第２の区分とを有する区分処理システムにおいてセキュリティを設ける方法であって、

ａ）ユーザによる許可要求を前記第２の区分内の前記セキュリティ・クライアントに送るステップと、

ｂ）前記許可要求を前記セキュリティ・クライアントから前記第１の区分内の前記共通セキュリティ・サーバに送るステップと、

ｃ）前記許可要求に対する第１の応答を前記第１の区分内の前記共通セキュリティ・サーバから前記第２の区分内の前記セキュリティ・クライアントに送るステップとを有し、

区分間の前記要求または前記第１の応答のうちのいずれか一方の前記送信が主記憶域を経由し、

ｄ）さらに、前記セキュリティ・クライアントから前記ユーザに第２の応答を送るステップとを含む方法。

【請求項２】ステップｂの許可要求の送信が、

ｂ１）前記第２の区分内で稼働する前記セキュリティ・クライアントが信号を送って前記第１の区分で稼働している第１のプログラムに前記第１の区分内のプロキシ・クライアントを始動させるステップと、

ｂ２）前記プロキシ・クライアントから前記第１の区分内の前記セキュリティ・サーバに前記要求を送るステップとをさらに含む、請求項１に記載の区分処理システムにおいてセキュリティを設ける方法。

【請求項３】ステップｂまたはステップｃのいずれかが、前記第１の区分と前記第２の区分との間で共用される主記憶域を使用するステップを含む、請求項１に記載の区分処理システムにおいてセキュリティを設ける方法。

【請求項４】ステップｂまたはステップｃのいずれかが、メモリ間データ移動によって前記第１の区分と前記第２の区分との間にリンクされた主記憶域を使用するステップを含む、請求項１に記載の区分処理システムにおいてセキュリティを設ける方法。

【請求項５】ステップｂが、前記セキュリティ・サーバ固有のインタフェースを使用した前記プロキシ・クライアントによるプログラム呼出しを含む、請求項１に記載の区分処理システムにおいてセキュリティを設ける方法。

【請求項６】ステップｃが、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第１の区分内で稼働している第１のプログラムを介して前記第１の応答を送るステップをさらに含む、請求項１に記載の区分処理システムにおいてセキュリティを設ける方法。

【請求項７】前記第２の区分内の前記セキュリティ・クライアントから前記ユーザに送られる前記第２の応答がプログラム処置である、請求項１に記載の区分処理においてセキュリティを設ける方法。

2

【請求項８】共通セキュリティ・サーバを含む第１の区分とセキュリティ・クライアントを有する第２の区分とを有するセキュリティを設ける区分処理システムであって、

ユーザによる許可要求を前記第２の区分内の前記セキュリティ・クライアントに送る手段と、

前記許可要求を前記セキュリティ・クライアントから前記第１の区分内の前記共通セキュリティ・サーバに送る手段と、

前記許可要求に対する第１の応答を前記第１の区分内の前記共通セキュリティ・サーバから前記第２の区分内の前記セキュリティ・クライアントに送る手段とを有し、

区分間の前記要求または前記第１の応答のうちのいずれか一方の前記送信が主記憶域を経由し、

さらに前記セキュリティ・クライアントから前記ユーザに第２の応答を送る手段とを含む区分処理システム。

【請求項９】前記許可要求を送る前記手段が、

前記第１の区分内で稼働し、プロキシ・クライアントを始動させる第１のプログラムと、

前記第２の区分内で稼働している前記セキュリティ・クライアントによって前記第１のプログラムに信号を送り、それによって前記第１の区分内の前記プロキシ・クライアントを始動させる手段と、

前記プロキシ・クライアントから前記第１の区分内の前記セキュリティ・サーバに前記要求を送る手段とをさらに含む、請求項８に記載の区分処理システム。

【請求項１０】前記主記憶域が前記第１の区分と前記第２の区分との間で共用される記憶域を含む、請求項８に記載の区分処理システム。

【請求項１１】前記第１の区分と前記第２の区分との間にリンクされた記憶域をさらに含み、前記セキュリティ・クライアントから前記許可要求を送る前記手段がメモリ間データ移動機構を含む、請求項８に記載の区分処理システム。

【請求項１２】前記プロキシ・クライアントから前記要求を送る前記手段が前記セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出しを送る手段を含む、請求項８に記載の区分処理システム。

【請求項１３】前記共通セキュリティ・サーバから前記許可要求に対する応答を送る前記手段が、前記第１の区分内で稼働し、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記応答を送る第１のプログラムをさらに含む、請求項８に記載の区分処理システム。

【請求項１４】第１の区分が共通セキュリティ・サーバを含み、第２の区分がセキュリティ・クライアントを含む区分処理システムにおいてセキュリティを設けるコン

ピュータ・プログラムであって、前記システムに、
ａ）ユーザによる許可要求を前記第２の区分内の前記セキュリティ・クライアントに送る手順と、
ｂ）前記許可要求を前記セキュリティ・クライアントから前記第１の区分内の前記共通セキュリティ・サーバに送る手順と、
ｃ）前記許可要求に対する第１の応答を前記第１の区分内の前記共通セキュリティ・サーバから前記第２の区分内の前記セキュリティ・クライアントに送るステップであって、区分間の前記要求または前記第１の応答のうちのいずれか一方の前記送信が主記憶域を経由する手順と、
ｄ）前記セキュリティ・クライアントから前記ユーザに第２の応答を送る手順とを実現させる、コンピュータ・プログラム。
【請求項１５】前記許可要求を送る手順が、
ｂ１）前記第２の区分内で稼働する前記セキュリティ・クライアントが信号を送って前記第１の区分で稼働している第１のプログラムに前記第１の区分内のプロキシ・クライアントを始動させる手順と、
ｂ２）前記プロキシ・クライアントから前記第１の区分内の前記セキュリティ・サーバに前記要求を送る手順とをさらに含む、請求項１４に記載のコンピュータ・プログラム。
【請求項１６】前記手順ｂまたは手順ｃにおいて、前記第１の区分と少なくとも１つの前記第２の区分の１つとの間で共用される記憶域を使用する、請求項１４に記載のコンピュータ・プログラム製品。
【請求項１７】前記手順ｂまたは手順ｃにおいて、前記第１の区分と少なくとも１つの前記第２の区分の１つとの間にメモリ棚データ移動機構によってリンクされた記憶域を使用する、請求項１４に記載のコンピュータ・プログラム製品。
【請求項１８】前記手順ｂにおいて、前記セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出しを供給する、請求項１４に記載のコンピュータ・プログラム製品。
【請求項１９】前記手順ｃにおいて、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第１の区分内で稼働している第１のプログラムを介して前記応答を送る、請求項１４に記載のコンピュータ・プログラム製品。
【請求項２０】共通セキュリティ・サーバを含む第１の区分とセキュリティ・クライアントを有する第２の区分とを有するセキュリティを設ける区分処理システムであって、
前記第１の区分によるアクセスが可能な第１の部分と前記第２の区分によるアクセスが可能な第２の部分とを有する主記憶域と、
前記セキュリティ・クライアントに接続され、ユーザによる許可要求を前記セキュリティ・クライアントに送る機構と、
前記セキュリティ・クライアントから前記共通セキュリティ・サーバに前記許可要求を送る第１の送信部と、
前記第１の区分内の前記共通セキュリティ・サーバから前記第２の区分内の前記セキュリティ・クライアントに前記許可要求に対する第１の応答を送る前記共通セキュリティ・サーバ内の第２の送信部と、を有し、前記区分間の前記要求または前記第１の応答のいずれか一方の送信が主記憶域を介し、
さらに、前記セキュリティ・クライアントから前記ユーザに第２の応答を送る前記共通セキュリティ・サーバ内の第３の送信部とを含む、区分処理システム。
【請求項２１】前記第１の送信部が、
プロキシ・クライアントを始動させる前記第１の区分内で稼働するプログラムと、
前記プログラムに信号を送り、それによって前記第１の区分内の前記プロキシ・クライアントを始動させる前記セキュリティ・クライアントと、
前記プロキシ・クライアントから前記セキュリティ・サーバに前記要求を送る第４の送信部とを含む、請求項２０に記載の区分処理システム。
【請求項２２】前記主記憶域が、前記第１の区分と前記第２の区分の両方によるアクセスが可能な第３の部分を含む、請求項２０に記載の区分処理システム。
【請求項２３】前記第１の区分と前記第２の区分の間にリンクされた記憶域をさらに含み、前記第２の送信部が前記主記憶域の前記第１の部分と前記第２の部分との間でデータを移動するメモリ間データ移動機構を含む、請求項２０に記載の区分処理システム。
【請求項２４】前記第４の送信部が、セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出を送る、請求項２１に記載の区分処理システム。
【請求項２５】前記第２の送信部が、前記第１の区分内で稼働し、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第１の応答を送るプログラムをさらに含む、請求項２０に記載の区分処理システム。
【請求項２６】　第２のセキュリティ・クライアントを有する第３の区分をさらに含み、前記第１の区分の前記共通セキュリティ・サーバが前記第２の区分内の前記セキュリティ・クライアントまたは前記第３の区分内の前記第２のセキュリティ・クライアントからの許可要求に応答する、請求項２０に記載の区分処理システム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、一般には区分データ処理システムに関し、例えば、システムの各区分内で複数のオペレーティング・システム・イメージを動作さ

5

せることが可能な単一プロセッサ・システムおよびマルチプロセッサ・システムに関する。複数のオペレーティング・システムの各オペレーティング・システムは、同種区分処理環境における同一のオペレーティング・システムのイメージとすることができるか、または異種区分処理環境において複数のオペレーティング・システムが複数のオペレーティング・システム・イメージによってサポートされる。

【０００２】

【従来の技術】最近の中企業から大企業までの企業のほとんどは、ＩＴ基盤を展開させて、従来集中化されていた「ガラス張り」データ・センターのカバー範囲を、組織全体、さらには組織の境界を超えて拡張している。このような展開の推進力は、一つには、従来ばらばらだった部門別業務を相互に結合し、供給業者および顧客とリアルタイムでコミュニケーションしたいという要求に根ざしており、電子商取引と、それに付随するそのような接続性をもたせるためにますます提供されるようになっている相互接続ソリューションおよび企業間ソリューションへのアクセスのための媒体としてのインターネットの急激な発展によって、さらに拍車がかけられている。

【０００３】この最近の展開に伴って、最近の企業は多くの異なるオペレーティング・プラットフォームを動的にリンクして、シームレスに相互接続されたシステムを構築することが必要になっている。企業は、合併活動から生じる非集中的購買業務、アプリケーション・ベースの要件、異種技術プラットフォームなどの要因のために、異種情報システムによって特徴づけられることが多い。さらに、供給業者、提携先、顧客の間のリアルタイムの企業外接続性を円滑にしたいという要求が、異種環境において接続性を持たせようという強い動機となっている。

【０００４】顧客要件の急激な増大に応えて、情報技術の分野ではそのようなニーズに対応して企業データ・センターのための接続性を拡張するデータ処理ソリューションが開発され始めている。

【０００５】本明細書の主題に関連する背景情報は例えば、複数区分のワークロード管理について記載されているＩＢＭ資料ＳＧ２４－５３２６－００「OS/390 WorkloadManager Implementation and Exploitation」(ISBM: 0738413070)、およびＥＳＡ／３９０命令セット・アーキテクチャについて記載されているＩＢＭ資料ＳＡ２２－７２０１－０６「ESA/390 Principles of Operation」などである。

【０００６】まず最初に、動作上の相互依存性があると考えられる様々なアプリケーションの処理サポートを同時に提供する統合システムを供給する必要のために、区分多重処理システムの市場が拡大している。単一の物理コンピューティング・システム内で複数のオペレーティング・システム・イメージをサポートする機能を備えた

6

この種の区分システムは、かつてはメインフレーム・コンピュータ（ＩＢＭＳ／３９０システムなど）の独壇場だったが、ますます広範囲な供給業者から販売されるようになってきている。たとえば、サン・マイクロシステムズ社は、最近、Ultra Enterprise 10000ハイエンド・サーバでシステム区分化の一形態を提供し始めた。これについては、米国特許第５９３１９３８号で詳述されている。他の会社も、このタイプのシステムに対する関心を示す方針書を出している。

【０００７】業界のこの動向は、企業内の様々な計算ワークロードを１つの（または少数の）物理サーバ・コンピュータに統合する際と、動的に再構成可能なハードウェア環境において試験レベルのコードと実働レベルのコードを同時に実施するための、システム区分化という「システム内システム」の利点を際だたせている。さらに、前掲の相互参照特許出願に記載されているＩＢＭＳ／３９０コンピュータ／システムなどの特定の区分多重処理システムでは、（プロセッサ、メモリ、入出力資源を含む）資源を、システム内で実行されるワークロードに割り当てられた区分に応じて、論理区分内および論理区分間で動的に割り振ることができる（ＩＢＭおよびＳ／３９０は、インターナショナル・ビジネス・マシーンズ・コーポレーションの登録商標である）。ワークロード優先順位に基づく動的資源割振りを可能にするこの機能は、従来からデータ・センター管理者がワークロードの一時的な急上昇に対応するために自分の予想計算ワークロードに意図的に余分な量の資源を割り当てる結果につながっている、長年の処理能力計画問題に対処するものである。

【０００８】

【発明が解決しようとする課題】これらの区分システムは、企業全体の異種システムを組み込むためのデータ・センターの拡張を容易にするが、現在、この種のソリューションは、異種または同種の区分プラットフォームを単一の相互運用区分システムに機能的に統合する単純な機構を提供しない。それどころか、この種の新型サーバは、オペレーティング・システム・イメージを単一の物理ハードウェア・プラットフォーム内に統合することはできるが、サーバの各区分内にあるオペレーティング・システム間の相互運用性の必要に十分に対処していない。この相互運用性の問題は、様々な区分内に異種オペレーティング・システムを有する異種システムではさらにひどくなる。さらに、この種のシステムは一般に、区分間の高帯域幅で低待ち時間の相互接続を可能にするタイプの異種プラットフォーム間の区分間資源共用に対応していなかった。このような相互運用性問題に対処することが重要な理由は、このような問題の解決策を組み込んだシステムがあれば、別個の区分で実行されているプロセス間の通信のためのより堅固な機構が可能になり、それによって、そのようなアプリケーションが別個のオ

ペレーティング・システム上で稼働しているがそれらのアプリケーションは実際には互いに対してローカルであるということを利用するようにすることができるためである。

【０００９】いくつかのオペレーティング・システムの「カーネル」の拡張機能によって、区分間メモリ共用を実施するための共用記憶域の使用を容易にすることが可能である。「カーネル」とは、オペレーティング・システム内の中核システム・サービス・コードである。このようにして形成された境界上でネットワーク・メッセージ受渡しプロトコルを実施することができるが、オペレーティング・システムのうちの１つまたは複数のオペレーティング・システムを修正する手段に訴えずに、効率的なプロセス間通信を可能にすることが望ましい場合が多い。また、米国特許第５９３１９３８号に記載されている米国サン・マイクロシステムズのＵｌｔｒａ　Ｅｎｔｅｒｐｒｉｓｅ１００００ハイ・エンド・サーバのようにメモリ領域を共用するために区分の分離を制限することを回避するのが望ましい場合が多い。それと同時に、区分間での情報の受渡しをネットワーク速度ではなくメモリ速度で行うことが望ましい。したがって、アドレスを共用せずに区分メモリ間で記憶内容を移動させる方法が望ましい。

【００１０】ＩＢＭ　Ｓ／３９０　Ｇｂｉｔ　Ｅｔｈｅｒｎｅｔ（Ｒ）（米国特許第５４４２８０２号）の入出力アダプタを使用して、１つの区分のカーネル・メモリから他の区分のカーネル・メモリにデータを移動させることができるが、データは第１カーネル・メモリからアダプタ上の待ち行列バッファに移動され、それからアダプタ上の第２の待ち行列バッファに転送された後、第２のカーネル・メモリに転送される。これは、メモリからメモリに転送するのに合計３回のデータ移動があることを意味する。どのようなメッセージ受渡し通信方式でも、データ・アクセスの待ち時間が共用記憶域との間での１回の記憶取出しの待ち時間に近づくように、データ移動操作の数を最小限にすることが望ましい。移動機能には、転送される各データ・ブロックごとに３回のデータ移動操作がある。これらの操作のうちの１つまたは２つをなくす方法が望ましい。

【００１１】同様に、ＩＢＭ　Ｓ／３９０並列シスプレックス結合機構機は、区分間メッセージ受渡しを容易にすることができ、そのために使用される。しかし、この場合、第１のカーネル・メモリから結合機構へ、次に結合機構から第２のカーネル・メモリにデータの転送が行われる。これは、望ましい１回の移動ではなく２回のデータ操作を必要とする。

【００１２】多くのコンピュータ・システムでは、無許可または不当なアクセスによるコンピュータ上のデータやアプリケーションの不正使用を防止するように、ユーザの識別を妥当性検査することが望ましい。様々なオペレーティング・システムおよびアプリケーション・システムが、そのためのユーザ認証およびその他のセキュリティ・サービスを備えている。区分システム、または実際にはどのようなシステムのクラスタまたはネットワークにであっても、アクセスするユーザをアクセス時点または重要な資源の要求や重要なシステム・メンテナンス機能の実行などの重要なチェックポイントで、一度妥当性検査することが望ましい。この要望を、「シングル・サイン・オン」要件と呼ぶ。このため、様々な区分のセキュリティ・サービスが相互作用するかまたは統合されなければならない。この例としては、ウェブから受け取った「デジタル認証」を処理し、それらをＯＳ／３９０内の従来のユーザＩＤおよびパスワード妥当性検査と権限授与にマッピングするＯＳ／３９０　ＳＡＦ（ＲＡＣＦ）インタフェースの拡張機能、ケルベロス・セキュリティ・サーバ、ディレクトリ・サービスのための最新のＬＤＡＰ標準などがある。

【００１３】さらに、電子商取引の競争的性質のため、ユーザ認証と権限授与は従来のシステムよりもさらに重要である。従業員であれば一日の始めに認証されるのを待つのを覚悟しているであろうが、顧客は認証にあまり時間がかかるとそのまま他の所に行ってしまうかもしれない。暗号の使用は、ウェブの公共的性質のために、この問題をさらに悪化させる。また、１つのオペレーティング・システム内に、他のオペレーティング・システムのためには作成されていないデバイス・ドライバが存在する場合も多い。そのような場合、１つの区分内のデバイス・ドライバに他の区分から効率的な方法でインタフェースすることが望ましい。現在、このタイプの動作のために利用可能な方法はネットワーク接続しかない。

【００１４】分散システムの問題の１つは、他のシステムが過剰に利用されている一方で、１つのシステム内の「ホワイト・スペース」または十分に利用されていない資源の管理である。システム間またはシステム・イメージ間で作業を移動する、ＩＢＭのLoadLeveler、またはＯＳ／３９０オペレーティング・システム・ワークロード・マネージャの並列シスプレックス機能などのワークロード・バランス機能がある。区分コンピューティング・システムでは、区分間で作業ではなく資源をシフトさせることが可能であり、望ましい。これが望ましい理由は、機能シフトに伴う大規模な環境切り換えとデータ移動が回避されるからである。

【００１５】シスプレックスの外部クラスタ化接続を使用してＵＮＩＸ（Ｒ）オペレーティング・システムのソケット間接続を実現するＩＢＭ　Ｓ／３９０のための「シスプレックス・ソケット」は、従来技術の一例である。この場合、サービスが、利用可能なセキュリティ・レベルを示し、必要なセキュリティ・レベルを示すアプリケーションの標識に基づいて接続を設定する。しかし、この場合、高いセキュリティ・レベルには暗号化が

設けられ、シスプレックス接続自体が、本発明によって実現されるメモリ接続よりもはるかに深い物理トランスポート層を有する。

【００１６】同様に、ＳＳＬ認証機能を備え、ウェブ・アプリケーション・サーバに（プロキシとして）認証情報を提供するウェブ・サーバは、本発明のメモリ共用または直接メモリ間メッセージを使用すれば有利なもう１つの例と見なすことができる。この場合、そのプロキシは、セキュリティ・サーバに渡すデータを再暗号化する必要がなく、さらに、管理すべき深い接続インタフェースもない。実際に、本発明のこの実施形態では、プロキシ・サーバは本質的に、セキュリティ・サーバと同じオペレーティング・システムの下で稼働するプロキシ・サーバと本質的に同じであるプロセスを介して、セキュリティ・サーバと通信することが、当業者ならわかるであろう。

【００１７】

【課題を解決するための手段】従来の技術の上述の問題および欠点は、本発明において克服されるとともに、本発明はその他の有利な特徴を備える。本発明の一態様は複数の異種オペレーティング・システム・イメージをサポートすることが可能な区分コンピュータ・システムであって、それらのオペレーティング・システム・イメージが、記憶場所を共用せずにメモリ内にあるそれらの記憶場所間でメッセージをメモリ速度で並列して受渡しすることができるコンピュータ・システムを含む。これは、１つの区分の１つのカーネル・メモリ空間から第２の区分のカーネル・メモリ空間へのデータの直接移動を容易にする、特別なデバイス・ドライバを備えた入出力アダプタを使用することによって行われる。

【００１８】開示の区分セキュリティ・システムは、共通セキュリティ・サーバを含む第１の区分と、セキュリティ・クライアントを有する第２の区分とを有する。区分処理システムはさらに、第１の区分がアクセス可能な第１の部分と、第２の区分がアクセス可能な第２の部分とを有する主記憶域を有する。また、セキュリティ・クライアントに接続され、ユーザがセキュリティ・クライアントに許可要求を送信するための機構も含まれる。セキュリティ・クライアント内の第１の送信部が、前記主記憶域を介してセキュリティ・クライアントから共通セキュリティ・サーバにこの許可要求を送る。共通セキュリティ・サーバ内の第２の送信部が、許可要求に対する応答を前記主記憶域を介して共通セキュリティ・サーバからセキュリティ・クライアントに送る。次に、セキュリティ・クライアント内の第３の送信部が、この応答をキュリティ・クライアントからユーザに送る。

【００１９】本発明の一実施形態では、共用メモリ資源は複数の区分内で実行される複数の相互運用プロセスのために、指定メモリ資源に独立してマップされる。このようにして、共通共用メモリ空間がメモリ資源を共用する各区分内のプロセスによってマップされ、区分内でそのプロセスに割り当てられ、通常のプロセス実行の課程でデータの読取りおよび書込みに使用可能なメモリ資源であるかのように見える。

【００２０】他の実施形態では、これらのプロセスは相互に依存し、共用メモリ資源は、いずれかのプロセスまたは両方のプロセスが後でアクセスするように、いずれかのプロセスまたは両方のプロセスから記憶することができる。

【００２１】本発明の他の実施形態では、システムは、各区分内の様々なプロセスを共用メモリ空間に接続するプロトコルを含む。

【００２２】本発明の他の実施形態では、１つの区分のカーネル空間から他の区分のカーネル空間へのデータの直接移動は、入出力アダプタによって可能になる。この入出力アダプタは、区分化に関わらず、すべての物理メモリに物理的にアクセスすることができる。入出力アダプタがすべてのメモリにアクセスすることができるのは、区分間での入出力資源共用を可能にする区分コンピュータ・システムにおける機能の自然の結果である。このような共用については、米国特許第５４１４８５１号に記載されている。しかし、この新規で発明的なアダプタは、データ移動機構を使用して１つの区分のメモリから別の区分のメモリに直接データを移動する機能を有する。

【００２３】本発明の他の実施形態では、カーネル・メモリ間でのデータの移動の機能は、ネットワーク通信アダプタのハードウェアとデバイス・ドライバ内で実施される。

【００２４】本発明の他の実施形態では、ネットワーク・アダプタは、ローカルであるがメモリ間インタフェースを介した異種セキュア接続のために最適化された、ＴＣＰ／ＩＰスタックによって駆動される。

【００２５】本発明の他の実施形態では、データ移動機構自体は、区分処理システムの通信ファブリックで実施され、さらに直接的なメモリ間転送を容易にする入出力アダプタによって制御される。

【００２６】本発明の他の実施形態では、データ移動機構は、特権ＣＩＳＣ命令のマイクロコードによって制御される。このマイクロコードは、オペランドとして与えられたネットワーク・アドレスおよびオフセットを物理アドレスに変換することができ、それによって、２つの区分内の実アドレスと仮想アドレスとを有する物理アドレス間の移動文字長命令に相当する機能を実行する（ＩＢＭ　Ｓ／３９０　ＭＶＣＬ命令。ＩＢＭ資料ＳＡ２２－７２０１－０６「ESA/390 Principles of Operation」参照）。

【００２７】本発明の他の実施形態では、データ移動機構は、ハイパバイザで実行されるルーチンによって制御される。このハイパバイザは、全物理メモリへの仮想メ

モリ・アクセスおよび実メモリ・アクセスを行うことができ、オペランドとして与えられたネットワーク・アドレスおよびオフセットを物理アドレスに変換することができ、それによって、２つの区分間の実アドレスと仮想アドレスとを有するアドレス間の移動文字長命令に相当する機能を実行する（ＩＢＭ　Ｓ／３９０　ＭＶＣＬ）。

【００２８】複数の区分のうちの１つの区分内のサーバ・プロセスとその他の区分内のクライアント・プロセスの実施により、区分システムは異種単一システム・クライアント・サーバ・ネットワークを実現することができる。既存のクライアント／サーバ・プロセスは一般に、ネットワーク・プロトコル接続によって相互動作するため、本発明のメッセージ受渡し実施形態で容易に実施され、インタフェース変更をせずにパフォーマンスおよびセキュリティ上の利点が得られる。しかし、本発明の共用メモリ実施形態でクライアント／サーバ・プロセスを実施すれば、配備のパフォーマンスまたは速度、あるいはその両方の点で有利になり得る。

【００２９】本発明の他の実施形態では、共用メモリまたはメモリ間メッセージ受渡しを使用するアプリケーション・サーバのために信用／保護サーバ環境が提供される。これによって、従来の技術のように追加の暗号化または認証を必要とせずに、許可および認証データの外面化というセキュリティ露出を回避することができる。

【００３０】本発明の特定の実施形態では、ウェブ・サーバは、ＯＳ／３９０、Ｚ／ＯＳ、またはＶＭ／３９０の下で稼働する「ＳＡＦ」セキュリティ・インタフェースへのメモリ・インタフェースを介して通信する、ＯＳ／３９０用Ｌｉｎｕｘ　Ａｐａｃｈｅである。この実施形態では、Ｌｉｎｕｘの「プラガブル・オーセンティケーション・モジュール（ＰＡＭ：Pluggable Authentication Module）」を修正して、メモリ接続を介してＳＡＦインタフェースを駆動する。

【００３１】本発明の他の実施形態では、セキュリティ証明／コンテキストが共用メモリに記憶されるか、またはメモリ間転送によって複製されるように、ポリシー・ディレクタやＲＡＣＦのようなセキュリティ・サーバに修正を加える。

【００３２】

【発明の実施の形態】本発明を構成するとみなされる主題は、本明細書の特許請求の範囲で具体的に示し、明確に請求する。本発明の上記およびその他の目的、特徴、および利点は、以下の詳細な説明を添付図面を参照しながら読めば明らかになろう。

【００３３】本発明の好ましい実施形態の特定の態様について説明する前に、区分処理システムの基本構成要素について説明すれば有益であろう。この説明を背景として用いれば、本発明の具体的な有利な特徴を区分システムでどのように使用して区分システムのパフォーマンスを向上させることができるかが、よりよく理解できよう。ＩＢＭ資料ＳＣ２８－１８５５－０６「OS/390 V2R7.0 OSA/SF User's Guide」を参照されたい。この資料には、ＯＳ／３９０オペレーティング・システムの一要素であるオープン・システムズ・アダプタ・サポート機構（ＯＳＡ／ＳＦ）の使用方法が記載されている。この資料は、ＯＳＡ／ＳＦのセットアップと、ＯＳ／２インタフェースまたはＯＳＡ／ＳＦコマンドのいずれかを使用してＯＳＡをカスタマイズし、管理するための解説書である。Ｇ３２１－５６４０－００「S/390 cluster technology: Parallel Sysplex」には、汎用大規模市場向けに開発されたクラスタ化マルチプロセッサ・システムについて記載されている。Ｓ／３９０並列シスプレックス・システムは、スケーラビリティの高いクラスタ化コンピュータ環境における完全データ共用および並列処理の利点を組み合わせるように設計されたアーキテクチャに基づく。並列シスプレックス・システムにより、コスト、パフォーマンス範囲、および可用性の分野において大きな利点が得られる。ＩＢＭ資料ＳＣ３４－５３４９－０１「MQSeries Queue Manager Clusters」には、MQSeries待ち行列管理プログラム／クラスタについて記載されており、クラスタの概念、用語、および利点について説明している。この資料には、新しいコマンドおよび変更されたコマンドの構文の概要が記載され、待ち行列管理プログラムのクラスタのセットアップおよびメンテナンスのための作業の例がいくつか示されている。ＩＢＭ資料ＳＡ２２－７２０１－０６「ESA/390 Principles of Operation」には、ＥＳＡ／３９０アーキテクチャの詳細な定義がリファレンスのために記載されている。この資料は、主としてアセンブラ言語プログラマ向けの解説書であり、各機能について、その機能に依存するアセンブラ言語プログラムを作成するのに必要な詳細度で記載されているが、ＥＳＡ／３９０の機能の詳細に関心のある人には有効であろう。

【００３４】上記の各資料には従来技術の例が示されており、本発明の背景を理解する上で有用であろう。

【００３５】図１を参照すると、区分処理システム１００を構成する基本要素が図示されている。システム１００は、ブロックＡおよびＢとして図示されているブロックに区分化することができる物理メモリ資源から成るメモリ資源ブロック１０１と、区分メモリ資源１０１と一致するように論理的または物理的に区分化することができる１つまたは複数のプロセッサで構成することができるプロセッサ資源ブロック１０２と、同様に区分化することができる入出力（Ｉ／Ｏ）資源ブロック１０３とから成る。これらの区分資源ブロックは、スイッチング・マトリックスなどを含むことができる相互接続ファブリック１０４を介して相互接続されている。相互接続ファブリック１０４は、プロセッサ１０２Ｂをメモリ１０１Ｂに接続するなどの区分内の資源を相互接続する機能を

果たすことができ、プロセッサ１０２Ａをメモリ１０１Ｂに接続するなど区分間の資源を相互接続する役割も果たすことができることを理解されたい。本明細書で使用する「ファブリック」という用語は、当技術分野で周知のシステムの各要素を相互接続する方法一般を意味することを意図したものである。これは単純なポイントツーポイント・バスであってもよく、高度なルーティング機構であってもよい。本発明の図面には２つの区分（ＡおよびＢ）を有するシステムを図示するが、このような図は本明細書の説明を簡単にするために選定したものであることと、本発明は、使用可能な資源と同じ数だけ、また区分化技法によって可能な数だけの区分が実現されるように構成可能なシステムを含むものと意図されていることが容易にわかるであろう。

【００３６】図の各区分ＡおよびＢは別々にして、別個のデータ処理システムの構成要素、すなわち、プロセッサ、メモリ、および入出力資源を構成することが容易にわかるであろう。このことは、区分処理システムに独自の「システム内システム」の利点を与える特徴である。実際に、本明細書で例示するように、現在使用可能な区分処理システム間の主な特徴は、システム資源を区分化することができる境界と、それらの境界を超えて区分間で資源を移動させることができる容易さである。

【００３７】区分を分離するが物理的境界である第１の事例の最も良い例は、サン・マイクロシステムズのUltra Enterprise 10000システムである。Ultra Enterprise 10000システムでは、区分は物理境界に沿って境界を画定され、具体的には、各システム・ボードがいくつかのプロセッサとメモリと入出力装置とを含む、１つまたは複数の物理的システム・ボードから成る領域または区分である。領域は、これらのシステム・ボードのうちの１つまたは複数のシステム・ボードおよびそれに接続された入出力アダプタであると定義される。各領域は、固有のバスとスイッチのアーキテクチャによって相互接続される。

【００３８】図２に、物理区分処理システム２００を構成する要素の高水準図を示す。図２を参照すればわかるように、システム２００は２つの領域または区分ＡおよびＢを含む。区分Ａは、２枚のシステム・ボード２００Ａ１および２００Ａ２から成る。区分Ａの各システム・ボードは、メモリ２０１Ａと、プロセッサ２０２Ａと、入出力装置２０３Ａと、相互接続媒体２０４Ａとを含む。相互接続媒体２０４Ａにより、システム・ボード２００Ａ１上の構成要素が互いに通信することができる。区分Ｂは単一のシステム・ボードから成り、同様の構成処理要素、すなわち、メモリ２０１Ｂ、プロセッサ２０２Ｂ、入出力装置２０３Ｂ、および相互接続媒体２０４Ｂを含む。区分にグループ分けされたこれらのシステム・ボードのほかに、各システム・ボードを結合し、１つの区分内のシステム・ボード間の相互接続と、異なる区分内のシステム・ボードの相互接続を可能にする相互接続ファブリック２０５がある。

【００３９】次のタイプのシステム区分は論理区分と呼ばれるものである。このシステムには、様々な区分への資源の割当を制約する物理境界がなく、システムは、物理的な場所には関係なくいずれの区分にでも割り当てることができる使用可能な資源のプールを有するものと見ることができる。たとえば、所与のシステム・ボード（システム・ボード２００Ａ１など）上のすべてのプロセッサが必ず同じ区分に割り当てられる物理区分システムとの違いである。ＩＢＭ　ＡＳ／４００システムが、論理区分専用資源処理システムの例である。ＡＳ／４００システムでは、ユーザが所与の区分にプロセッサ、メモリ、および入出力装置をそれらの物理場所に関係なく組み込むことができる。したがって、たとえば物理的に同じカード上にある２つのプロセッサを、２つの異なる区分のための資源として指定することができる。同様に、カードなどの所与の物理パッケージ内のメモリ資源は、そのアドレス空間の一部を論理的に１つの区分専用とし、残りの部分を別の区分専用とすることができる。

【００４０】ＡＳ／４００システムなどの論理区分専用資源システムの特徴は、資源の区分への論理マッピングが、システムの手動再構成によってのみ変更することができる静的に行われる割当てであることである。図３を参照すると、プロセッサ３０２Ａ１は、システム内の任意の場所に物理的に配置することができ、論理的に区分Ａ専用化されたプロセッサである。ユーザがプロセッサ３０２Ａ１を区分Ｂにマップし直したい場合、そのプロセッサをオフラインにし、その変更に対応するように手動でマップし直す必要がある。論理区分システムは、たとえば固定数のプロセッサをサポートするシステム・ボードなどの物理的区分化境界という制限による制約を受けないため、資源区分化の細分性が高い。しかし、このような論理区分専用資源システムの再構成は、区分再マッピングの対象となる資源の動作を中断させなければ行うことができない。したがって、このようなシステムは、物理区分システムに固有の制限のいくつかを回避するが、区分間での資源の静的マッピングに付随する再構成上の制約があることがわかる。

【００４１】このため、発明人等は、論理区分共用資源システムを考える。このようなシステムの一例は、ＩＢＭ　Ｓ／３９０コンピュータ／システムである。論理区分共用資源システムの特徴は、プロセッサなどの論理区分資源を複数の区分が共用することができることである。この特徴により、論理区分専用資源システムの再構成上の制約が事実上克服される。

【００４２】図４に、論理区分資源共用システム４００の全体構成を示す。論理区分専用資源システム３００と同様、システム４００は、システム内の物理的場所に関係なくいずれの区分（この例ではＡまたはＢ）にでも論

理的に割り当てることができるメモリ４０１と、プロセッサ４０２と、入出力資源４０３とを含む。しかし、システム４００でわかるように、特定のプロセッサ４０２または入出力資源４０３の論理区分割当ては、「ハイパバイザ」（４０８）で稼働しているスケジューラに従って仮想プロセッサ（４０６）および入出力ドライバ（４０７）をスワップすることによって動的に変更することができる。（ハイパバイザとは、仮想機械のために資源のスケジューリングと割振りを行う監視プログラムである。）プロセッサと入出力資源の仮想化により、区分をそれらの資源間で動的に共用することができるようにする適切な優先順位づけを使用して、全オペレーティング・システム・イメージを動作停止中にスワップすることができる。

【００４３】論理区分共用資源システム４００は、プロセッサおよび入出力資源を共用するための機構を備えるが、既存のシステムは区分間メッセージ受渡しに完全には対処していなかった。これは、既存の区分システムは区分間の通信を可能にすることができないということではない。実際には、本明細書に記載のように各タイプの区分システムではこのような通信が行われる。しかし、これらの実施態様のいずれも、ハイパバイザ、共用メモリ実施態様、または区分間を接続する標準アダプタまたはチャネル通信装置のセットまたはネットワークの介入なしにカーネル・メモリからカーネル・メモリにデータを移動する手段を備えていない。

【００４４】米国特許第５９３１９３８号に記載されているようなサン・マイクロシステムズのUltra Enterprise 10000に代表される物理区分多重処理システムでは、マスク／レジスタを適切に設定することによって、システム・メモリの一領域にハードウェア・レベルで複数の区分がアクセスすることができる。サンの特許では、区分間ネットワークのためのバッファ機構および通信手段として使用することができると記載している以外には、この機能の利用法を教示していない。

【００４５】「Coupling Facility Configuration Options: A Positioning Paper」（ＧＦ２２－５０４２－００、ＩＢＭコーポレーション）で詳述されているように、ＩＢＭ　Ｓ／３９０システムでは、共通にアドレス指定された物理メモリを「統合結合機構」として使用する同様の内部クラスタ化機能について記載されている。この場合、共用記憶域は実際にリポジトリであるが、この共用記憶域への接続はＸＣＦと呼ぶデバイス・ドライバのような入出力資源を介する。この場合、共用メモリは結合機構で実現されるが、非Ｓ／３９０オペレーティング・システムはそれを使用する拡張部を作成する必要がある。さらに、この実施態様では、データを１つの区分のカーネル・メモリから結合機構のメモリに移動させ、次に第２の区分のカーネル・メモリに移動させる。

【００４６】カーネルとは、オペレーティング・システムのうち、ハードウェア資源の割振りなどの基本機能を実行する部分である。カーネル・メモリとは、カーネルがその機能を実行するために使用する、カーネルが使用可能なメモリ空間である。

【００４７】それに対して、本発明は、区分またはハードウェア内にオペレーティング・システムの共用記憶域拡張部を設けずに、新しい入出力アダプタおよびそのデバイス・ドライバを使用可能にする機能を使用して１回の動作で１つの区分のカーネル・メモリから別の区分のカーネル・メモリにデータを移動する手段を設ける。

【００４８】本発明がどのように実現されるかを理解するには、オペレーティング・システムにおけるプロセス間通信について理解すれば役に立つ。図５を参照すると、プロセスＡ（５０１）およびＢ（５０３）がそれぞれ、アドレス空間であるメモリＡ（５０２）およびメモリＢ（５０４）を有する。これらのアドレス空間は、カーネル（５０５）によるシステム呼出の実行によってそれらに割り振られる実メモリを有する。カーネルは、それ自体のアドレス空間であるメモリＫ（５０６）を有する。通信の一形態では、プロセスＡとＢは、バッファ５１０の作成、接続、アクセスを行う適切なシステム呼出しを行うことにより、メモリＫ内にバッファ５１０を作成することによって通信する。これらの呼出のセマンティクスはシステムによって異なるが、その効果は同じである。第２の通信形態では、メモリＳ（５０７）のセグメント５１１を、メモリＡ（５０２）およびメモリＢ（５０４）のアドレス空間にマップする。このマッピングが完了した後は、プロセスＡ（５０１）およびＢ（５０３）は、両方のプロセスが認識する任意のプロトコルに従ってメモリＳ（５０７）の共用セグメントを自由に使用することができる。

【００４９】図６で、プロセスＡ（６０１）とプロセスＢ（６０３）とが異なるオペレーティング・システム・ドメイン、イメージ、または区分（区分１（６１４）および区分２（６１５））にある。ここでは、カーネル・メモリとしてメモリＫ１（６０６）およびメモリＫ２（６０８）を有するカーネル１（６０５）およびカーネル２（６０７）がある。この場合、メモリＳ（６０９）は、区分１と区分２の両方がアクセスすることができる物理メモリの空間である。このような共用は、ＵＥ１０００００メモリ・マッピング実施態様またはＳ／３９０ハイパバイザ実施態様などを限定せずに任意の実施態様に従って、または区分化によって作成されるアクセス障壁を限定するその他の手段に従って、使用可能化することができる。他の例として、共用空間を定義する構成レジスタ内に先頭アドレスのある、最上位物理メモリ・アドレスに共用メモリをマップする。

【００５０】規則として、メモリＳ（６０９）は、メモリＫ１およびメモリＫ２にマップされるカーネル１およびカーネル２の拡張部によって使用される共用セグメン

ト（６１０）を有する。セグメント６１０を使用して、メモリＫ１（６０６）およびメモリＫ２（６０８）にマップされて、前述の第１の形態に従って区分間通信を可能にする、メモリ（６０９）のセグメントのための定義および割振りテーブルを保持するか、または、図５を参照しながら前述した第２の通信形態に従ってメモリＡ（６０２）およびメモリＢ（６０４）にマップされるセグメントＳ２（６１１）を定義する。本発明の一実施形態では、メモリＳはサイズが限定されており、実記憶域に固定される。しかし、付随するページ管理タスクが効率的に管理される限り、メモリは固定される必要はなく、それによってより大きな共用記憶空間が可能になるものと企図される。

【００５１】本発明の第１の実施形態では、共用記憶域の定義および割振りテーブルは、共用メモリ構成データ・セット（ＳＭＣＤＳ）（６１３）からデータを読み取り、メモリＳ（６０９）のセグメントＳ（６１０）内にテーブルを作成する共用メモリ構成プログラム（ＳＭＣＰ）（６１２）と呼ばれるスタンドアロン・ユーティリティ・プログラムによってメモリ内に設定される。したがって、どのカーネルが記憶域のどのセグメントを共用するかの割振りと定義は固定されており、このユーティリティによって作成された構成によって事前決定される。その後、様々なカーネル拡張部がその共用記憶域を使用して、パイプ、メッセージ待ち行列、ソケット、さらには、一部のセグメントをそれ独自の規則に従って共用メモリ・セグメントとしてユーザ・プロセスに割り振ることなど、様々なイメージ間、プロセス間通信構成物を実現する。このようなプロセス間通信は、ＩＰＣ ＡＰＩ６１８および６１９を介して使用可能にされる。

【００５２】共用記憶域の割振りテーブルには、イメージ識別子、セグメント番号、グループＩＤ、ユーザＩＤ、「スティッキィ・ビット」、および許可ビットから成る項目が含まれる。スティッキィ・ビットは、関連する記憶域がページ可能でないことを示す。この例示の実施形態では、スティッキィ・ビットは予約済みであり、１をとる（すなわち、データはメモリ内のその場所に固定すなわち「スティック」される）。セグメントを使用する各グループ、ユーザ、およびイメージは、このテーブル内に項目を持つ。規則として、すべてのカーネルがテーブルを読み取ることができるが、どのカーネルもテーブルに書き込むことはできない。初期設定時に、カーネル拡張部は構成テーブルを読み取り、他のプロセスによってイメージ間プロセス間通信が要求されたときに使用するためにそれ独自の割振りテーブルを作成する。カーネルは、割り振られた空間の一部または全部を、プロセス間通信を要求する他のプロセスの要求に作成する「パイプ」、ファイル、およびメッセージ待ち行列の実現のために使用する。パイプは、１つのプロセスからカーネル機能を介して第２のプロセスに向けて送られるデータである。パイプ、ファイル、イメージ待ち行列は、Ｌｉｎｕｘ、ＯＳ／３９０ＵＳＳ、およびほとんどのＵＮＩＸ（Ｒ）オペレーティング・システムで使用されているような、標準ＵＮＩＸ（Ｒ）オペレーティング・システムのプロセス間通信ＡＰＩおよびデータ構造である。共用空間の一部は、直接クロス・システム・メモリ共用のために、さらに他のカーネル拡張部によって他のプロセスのアドレス空間にマップすることができる。

【００５３】共用メモリの割振り、使用、および仮想アドレス空間へのマッピングは、各カーネルがそれ自体の規則と変換プロセスとに従って行うが、基本的なハードウェア・ロックとメモリ共用プロトコルは、システムの他の部分の基礎にある共通ハードウェア設計アーキテクチャによって駆動される。

【００５４】より高水準のプロトコルは、通信が行われるためには共通でなければならない。好ましい実施形態では、これは、ＵＮＩＸ（Ｒ）オペレーティング・システムと共に使用するために、要求をクロス・イメージであると識別する拡張部分を付けて、様々なオペレーティング・システム・イメージのそれぞれにＩＰＣ（プロセス間通信）ＡＰＩを実施させることによって行われる。この拡張部分は、パラメータによって、または別個の新しい識別子／コマンド名によって行うことができる。

【００５５】図４および図７を参照すると、本発明は、チャネルまたはネットワーク接続を介したデータの転送と、オペレーティング・システムの共用メモリ拡張部の使用の両方を回避することがわかる。区分７１４内のアプリケーション・プロセス（７０１）が、ソケット・インタフェース７０８にアクセスし、ソケット・インタフェース７０８がカーネル１（７０５）を呼び出す。ソケット・インタフェースは、ＴＣＰ／ＩＰスタックの特定のポートをリスト・ユーザ・プロセスに関係づける構造体である。カーネルは、デバイス・ドライバ（７１６）にアクセスし、デバイス・ドライバ７１６は、入出力アダプタ（７２０）のハードウェアを介してカーネル・メモリ１（７０６）からカーネル・メモリ２（７０８）に、メモリ（４０１）にとってメモリ間移動のように見えるようにデータを転送し、区分７１４および７１５のプロセッサ（４０２）またはファブリック（４０４）あるいはその両方で実現されたキャッシュ・メモリを迂回する。データを移動した後、入出力アダプタは区分７１５内のデバイス・ドライバ（７１７）にアクセスし、データが移動されたことを示す。デバイス・ドライバ７１７は、次に、カーネル２（７０７）に、ソケット（７１９）にそのソケットを待っているデータがあることを示す。次に、ソケット（７１９）がそのデータをプロセス（７０３）に示す。したがって、直接メモリ間移動が行われると同時に、外部インタフェース上のデータの移動が回避され、メモリ共用のためのいずれのオペレーティング・システムの拡張も回避される。

【００５６】それに対して、図８に示す従来技術のシステムは、別々のメモリ移動動作を使用してカーネル・メモリ１（７０６）からアダプタ・メモリ・バッファ１（７２１）に移動する。第２のメモリ移動動作によって、データをアダプタ・メモリ・バッファ１（７２１）からアダプタ・メモリ・バッファ２（７２２）に移動する。次に、第３のメモリ移動動作によって、アダプタ・メモリ・バッファ２（７２２）からカーネル・メモリ２（７０８）にデータを移動する。これは、３回の別個のメモリ移動動作を使用して２つのカーネル・メモリ間でデータを移動することを意味する。それに対して、図７の本発明では、１回のメモリ移動動作で、データがカーネル・メモリ１（７０６）とカーネル・メモリ２（７０８）の間で直接移動する。これは、ユーザ・プロセスから見たときに待ち時間を短縮する効果がある。

【００５７】本発明の他の実施形態を図４および図９に示す。ここでは、ファブリック（４０４）で実データ移動機構ハードウェアが実現される。この実施形態の動作は、上記の説明のように進むが、入出力アダプタ８２０内の制御状態（８２２）に従って、データが実際にはファブリック（４０４）内の移動機構ハードウェアによって移動される点が異なる。

【００５８】このようなファブリックに配置されたデータ移動機構の一例は、米国特許第５２６９００９号に記載されている。この参照特許に記載されている機構は、区分の主記憶域場所間でのデータ転送を含むように拡張されている。

【００５９】実施形態を問わず、本発明は以下の要素を含む。すなわち、ＣＰＵの設計によって規定された基礎共通データ移動プロトコルと、入出力アダプタまたはファブリックあるいはその両方のハードウェアと、入出力アダプタへのインタフェースを実現する異種セット・デバイス・ドライバと、本実施形態ではソケット・インタフェースとして示されている共通高水準ネットワーク・プロトコルと、ネットワーク・アドレスを入出力アダプタ（８２０）が各区分のカーネル・メモリおよびデバイス・ドライバと通信するために使用する物理メモリ・アドレスおよび入出力割込みベクトルまたはポインタにマッピングすることとである。

【００６０】データ移動機構は、ハードウェア状態機械として入出力アダプタ内で実現するか、またはマイクロコードとマイクロプロセッサとで実現することができる。あるいは、入出力アダプタによって制御される、機械の通信ファブリック内のデータ移動機構を使用するなどで実現することができる。このようなデータ移動機構の一例は、米国特許第５２６９００９号に記載されている。

【００６１】図１０を参照すると、実施態様を問わず、データ移動機構は以下の要素を有する。メモリからのデータがソース・レジスタ（９０１）に保持され、そのデータがデータ・アライナ（９０２および９０４）を介して宛先レジスタ（９０３）に渡され次にメモリへ返される。したがって、メモリ・フェッチがあり、その後でメモリ・ストアが、連続動作の一部としてある。すなわち、メモリ・ラインから複数のワードがフェッチされると、位置合わせプロセスが行われる。位置合わせされたデータは、メモリ・ストアが開始されるまで宛先レジスタ（９０３）にバッファリングされる。ソース・レジスタ（９０１）と宛先レジスタ（９０３）を使用して、移動動作中にフェッチとストアの間でどの程度の重なり合いが可能かに応じてメモリ・データの単一のラインまたは複数のラインを保持することができる。メモリのアドレス指定は、移動中にフェッチ・アドレスとストア・アドレスを追跡するカウンタ（９０５および９０６）によって行われる。制御およびバイト・カウント要素（９０８）が、アライナ（９０２および９０４）を介したデータの流れを制御し、メモリ・アドレスに対するソース・カウンタ（９０５）または宛先カウンタ（９０６）の選択（９０７）が行われるようにする。このコントローラ（９０８）は、アドレス・カウンタ（９０５および９０６）の更新も制御する。

【００６２】図１１を参照すると、データ移動機構は、デバイス・ドライバによって実施される特権ＣＩＳＣ命令（１０００）として実現することもできる。このようなＣＩＳＣ命令は、Ｓ／３９０ページ移動、、文字長移動などの区分内データ移動にハードウェア機能を利用するが、テーブル・マッピング・ネットワーク・アドレスと、物理メモリ・アドレスまでのオフセットに従って、物理的にメモリをアドレス指定する特権も有することになる。最後に、データ移動機構およびアダプタは、仮想アダプタとして機能するハイパバイザ・コードによって実現することができる。

【００６３】図１２に、データ移動機構がアダプタ内にある場合の、データ移動機構の以下のステップから成る動作を示す。

１１０１　ユーザが、以下を供給するデバイス・ドライバを呼び出す。
　　ソース・ネットワークＩＤ
　　ソース・オフセット
　　宛先ネットワークＩＤ
１１０２　デバイス・ドライバがアドレスをアダプタに転送する
１１０３　アダプタがアドレスを変換する
　　ＩＤから物理基底アドレスを探索する（テーブル・ルックアップ）

ロックおよび現行宛先オフセットを入手する
オフセットを加える
境界を調べる
１１０４　アダプタがカウントとアドレスをレジスタにロードする
１１０５　アダプタがデータ移動を実行する
１１０６　アダプタがロックを解放する
１１０７　アダプタがデバイス・ドライバに通知し、デバイス・ドライバがユーザに「戻る」

きるプロセッサ通信ファブリック内で実施されるデータ移動方法を示す。

【００６４】図１３に、以下の方法を使用することがで

１２０１　ユーザが、以下を供給するデバイス・ドライバを呼び出す。
ソース・ネットワークＩＤ
ソース・オフセット
宛先ネットワークＩＤ
１２０２　デバイス・ドライバがアダプタにアドレスを送る
１２０３　アダプタがアドレスを変換する
ＩＤから物理基底アドレスを探索する（テーブル・ルックアップ）
ロックおよび現行宛先オフセットを入手する
オフセットを加える
境界を調べる
アダプタがロックと物理アドレスをデバイス・ドライバに返す
１２０４　デバイス・ドライバがデータ移動を実行する
１２０５　デバイス・ドライバがロックを解放する
１２０６　デバイス・ドライバが戻る

【００６５】以上、区分コンピュータ・システムにおいて異種相互動作を実現する２通りの方法について述べた。一方は、共用メモリ機構とオペレーティング・システム・カーネルの拡張部とを使用して、区分間プロセス間通信プロトコルを使用可能にし、他方は共用入出力アダプタの機能を仕様して、全物理メモリをアドレス指定し、１回の動作でメモリ間メッセージ受渡しを実現する。

【００６６】以上の構成は、単一システム・クライアント／サーバ・モデルを利用したいくつかの新規な実施態様を生み出す。この構成を実現する１つの方法は、サーバ作業待ち行列を共用記憶空間に入れ、様々なクライアントが要求を付加することができるようにすることである。その場合、「リモート」クライアントのための戻りバッファも共用メモリ空間に入れ、それによってクライアントがその中に入っている情報にアクセスすることができるようにしなければならない。あるいは、前述のメッセージ受渡し方式を使用して、既存のネットワーク指向クライアント／サーバを迅速かつ容易に配備することができる。これらの実施態様は、例示として示したものであり、新規かつ発明的であるが、限定的なものとみなしてはならない。実際に、当業者なら、この構成を基にして様々な方法で単一のシステム・パラダイム内で異なるタイプの異種クライアント／サーバ・システムを実現することができ、実現するであろうことが容易にわかるであろう。

【００６７】区分のクラスタのワークロード管理

図１４を参照すると、ＯＳ／３９０オペレーティング・システムのワークロード・マネージャ（ＷＬＭ）（１３０８）は、Ｓ／３９０の区分ハイパバイザと通信して、各区分の割り振られた資源を調整することができる。これはＬＰＡＲクラスタ化と呼ばれる。しかし、非ＯＳ／３９０区分（１３０１）の場合、ＷＬＭは、使用率と、ハイパーバイザが供給することができるその他の情報だけに基づいて割振りを行わなければならず、区分のオペレーティング・システムまたはアプリケーションには基づかない。上記の低待ち時間区分間通信（１３０５）を使用して、情報を区分からＷＬＭ（１３０８）にパイプすれば、ＷＬＭ（１３０８）にシステム資源間割振りのより効果的な作業を行うのに必要な情報を提供するきわめてオーバーヘッドの少ない手段となる。これは、アプリケーションがワークロード管理の装備を備えていない場合であっても有効になり得る。その理由は、制御されるシステムは一般に、システムを出入りするＩＰパケットをカウントするＴＣＰ／ＩＰスタック内のパケット活動カウンタにアクセスするコマンド、ＵＮＩＸ（Ｒ）オペレーティング・システムの「ＮＥＴＳＴＡＴ」（ＵＮＩＸ（Ｒ）オペレーティング・システム標準コマンド・ライブラリの一部）を実施することができ、また、使用中と遊休状態のサイクルをカウントし、使用率データ（１３０２）を生成するカーネル内のシステム活動カウンタにアクセスする標準ＵＮＩＸ（Ｒ）オペレーティン

グ・システム・コマンドＵＮＩＸ(R)オペレーティング・システムの「ＶＭＳＴＡＴ」（ＵＮＩＸ（Ｒ）オペレーティング・システム標準コマンド・ライブラリの一部）も実行することができるためである。必ずしも既存のＮＥＴＳＴＡＴコマンドおよびＶＭＳＴＡＴコマンドを使用する必要はなく、パケット・カウントと使用率を供給する基礎機構を使用して資源とパス長のコストを最小限にするのが最もよいということがわかるであろう。このデータを「速度」メトリック（１３０３）に統合し、それをワークロード・マネージャ（ＷＬＭ）区分（１３０７）に送ることによって、ＷＬＭ（１３０８）はハイパーバイザに資源の調整を行わせることができる。ＣＰＵ使用率が高く、パケット・トラフィックが低い場合、その区分はより多くの資源を必要とする。接続（１３０４および１３０６）は、相互接続（１３０５）の実施形態によって異なる。共用メモリ実施形態では、これらの接続はＵＮＩＸ（Ｒ）オペレーティング・システムのＰＩＰＥ、メッセージＱ、ＳＨＭＥＭ、またはソケット構成とすることができる。データ移動機構実施形態では、これらは一般には、ソケット接続となる。

【００６８】本発明の一実施形態では、「速度」メトリックは、以下のようにして得られる（ＩＢＭレッドブック資料番号ＳＧ２４－４８１０－０１「Understanding RS/6000 Performance and sizing」に記載されているＵＮＩＸ（Ｒ）オペレーティング・システム・コマンドＮＥＴＳＴＡＴおよびＶＭＳＴＡＴを参照）。

（ＮＥＴＳＴＡＴ）合計パケット数のインターバル・データを使用してスループットを求める。インターバルＣＰＵデータ（ＶＭＳＴＡＴ）を使用してＣＰＵ使用率を求める。これらのデータをプロットし、ピークを１として正規化したトラフィックと共に表示する（１４０１）。トラフィックとＣＰＵとの累積相関分析を行う（１４０２）。トラフィックの関係を関数Ｔ（Ｃ）に曲線当てはめする。この例（１４０２）では、Ｔ（Ｃ）＝０．８６４＋１．１２Ｃである。Ｓ＝ｄＴ／ｄＣが速度メトリックであり、この例ではＳ＝１．１２である。Ｓが傾向線よりも小さい場合、資源が必要とされる。

【００６９】図１５の例では、これを２回（１４０３および１４０４）行う。管理図が、業界で監視プロセスを作成する標準的な方法である。１４０５の管理図のようにＳを動的にプロットする。上記でパケット・トラフィックとＣＰＵとの関係について見たような関係を考えると、収集したデータを統計的管理理論に基づいて様々な方法で監視し、配列することができる。これらの方法は一般に、アクションを引き起こす、制御変数のしきい値に依存する。あらゆるフィードバック・システムと同様に、管理不能状態に近いと判断されたらただちにアクションを起こす必要がある。そうしないとシステムが不安定になる可能性がある。本発明では、これは、内部通信による低待ち時間接続によって行われる。

【００７０】静的環境では、Ｓを使用して、より多くの資源が必要になる使用率を設定することができる。これが機能する限り、平均を超えるＳもワークロードと時間との関数である。図１５を参照すると、まず、これは５０％と６０％との間にあるように見えることがわかり、次に、Ｓの谷に続いて少なくとも１時間間隔で使用率のピークが現れることがわかる。したがって、Ｓは資源のより適時な調整を可能にする「先行標識」であるため、ＷＬＭは使用率ではなくＳを供給した方がより効果的である。区分計算機の資源は区分間で共用されるため、ワークロード・マネージャはこのＳデータを複数の区分から入手しなければならない。きわめて低いオーバーヘッドおよび高速でデータの転送を行う必要がある。本発明は、この両方の条件を可能にする。図１４を参照すると、ワークロード・マネージャのない区分（１３０１）で、モニタが使用率とパケット・データを監視し（１３０２）、それがプログラム・ステップで使用され（１３０３）、パラメータ（この例ではＳ）が評価される。次にプログラムは低待ち時間区分間通信機構（１３０５）への接続（１３０４）を使用し、低待ち時間区分間通信機構はワークロード・マネージャがある区分（１３０７）内の接続（１３０６）にそれを渡し、これが「論理区分クラスタ・マネージャ」（１３０８）に接続して入力データを渡す。論理区分クラスタ・マネージャについては、米国特許第０９／６７７３３８号に記載されている。

【００７１】この場合、区分データをワークロード・マネージャに伝える最も効率的な方法は、メモリ共用によるものであるが、ソケット待ち時間がデータの送達時間の余裕を持たせることができるほど低い場合には、内部ソケット接続も有効である。これは、ワークロードと、必要な制御細分性の両方に依存する。

【００７２】上記は、ワークロード・マネージャが資源を割り振るための情報を供給する新規で発明的な方法であるが、決して限定的なものととってはならない。この例は、多くの新しいコードを使用せずに、すべてではないがほとんどのオペレーティング・システムから得られるメトリックであるため選定している。クライアント・システムは、応答時間やユーザ数など、ＷＬＭサーバに渡す任意のメトリックの任意の計測を実施することができる。

【００７３】間接入出力デバイス・ドライバが、ハードウェアによってサポートされている可能なオペレーティング・システムのうちの１つのオペレーティング・システムのみで使用可能な場合がある。共用メモリにデバイス・ドライバ・メモリ・インタフェースを渡し、すべての接続システムがドライバ・プロトコルを遵守することによって、装置を複数のシステムで共用することができる。実際には、１つの区分が他の区分のためのＩＯＰになることができる。装置へのアクセスは、装置を過負荷

にすると単一システムから過負荷にするのと同じ悪い結果になるという認識のもとで、単一システム・レベルに近い。図１６を参照すると、デバイス・ドライバ（１５０１）は、共用メモリ（１５１１）を介してアプリケーションおよびアクセス方法（１５０３）からの入出力サービス要求に応答する。

【００７４】メッセージ受渡し実施態様はある種の装置には使用することができるが、ソケット、スタック、およびデータ移動の待ち時間を受容しなければならないであろう。これは、ネイティブ装置とネットワーク接続装置の間と見ることができる。

【００７５】デバイス・ドライバを稼働させているシステム・イメージに割り振られたプロセッサ資源を、アプリケーションを稼働させているシステム・イメージに割り振られたプロセッサ資源から分離すれば、さらなる機能強化が得られる。これを行った場合、入出力割込みの対象ではないプロセッサにおける、入出力割込みおよびそれに付随するコンテキスト切り換えによるキャッシュおよびプログラムの流れの中断が回避される。

【００７６】共通セキュリティ・サーバ

アプリケーションがウェブ対応になり、統合化されるにつれて、ユーザの検証と権限の設定が従来のシステムよりも浸透するようになる。この計算が、異種システムをまとめてアプリケーションを統合するのに必要である。その結果、ＬＤＡＰ、ケルベロス、ＲＡＣＦ、およびその他のセキュリティ機能を統合された方式で使用するには、通常、セキュリティ機能を実行するために共通セキュリティ・サーバへのネットワーク接続が必要である。これはパフォーマンスに影響する。また、ネットワーク・スニファのセキュリティ露出もある。共通セキュリティ・サーバを共用メモリ接続またはメモリ移動機構接続を介してウェブ・サーバに接続すれば、この活動を大幅に高速化することができ、接続が内面化され、セキュリティが向上する。さらに、このような環境では、他のＵＮＩＸ（Ｒ）オペレーティング・システムよりも、パスワード保護に基づくＳ／３９０「ＲＡＣＦ」またはその他のＯＳ／３９０「ＳＡＦ」インタフェース・ユーザ認証の強化されたセキュリティの方を選ぶ顧客もいると考えられ、ＬＩＮＵＸの場合には特にそうである。Ｌｉｎｕｘシステムでは、そのような共用サーバのためのクライアント側の構築が比較的容易であるが、それはユーザ認証が、調整とカスタマイズが行われることを意図した「プラガブル・オーセンティケーション・モジュール（Pluggable authentication module）」によって行われるためである。この場合、セキュリティ・サーバには共用メモリ・インタフェースまたはメモリ間データ移動機構インタフェースを介してアクセスし、ウェブ・サーバがこのインタフェースを争奪する。その結果の作業待ち行列は、セキュリティ・サーバが共用メモリ・インタフェースを介して必要に応じて応答することによって稼働する。その結果、ウェブ・アプリケーションのセキュリティとパフォーマンスが向上する。図１７を参照すると、セキュリティ・サーバ（１６０１）は、共用メモリ（１６１１）を介してユーザ・プロセス（１６０３）からのアクセス要求に応答する。ユーザ・プロセスは、カーネル２（１６０７）内のセキュリティ・クライアント・プロセス（これはＬＩＮＵＸの場合のＰＡＭである）への標準インターネット・プロセス通信（ＩＰＣ）インタフェースを使用し、次にこれが共用メモリ（１６１０）を介してカーネル１（１６０５）内のカーネル・プロセスに連絡し、さらにこのカーネル・プロセスが、ユーザ・プロセス（１６０３）のプロキシとしてセキュリティ・サーバ・インタフェース（ＯＳ／３９０またはＺ／ＯＳの場合のＳＡＦ）を駆動し、共用メモリ（１６１０）を介してカーネル２（１６０７）内のセキュリティ・クライアントに許可を返す。

【００７７】本発明は、データを自由に流したり、追加の暗号化や認証を必要とするものよりもはるかに安全な共用メモリを使用して、アプリケーション・サーバに提供可能な信用／保護環境を向上させる。

【００７８】本発明は多くの改良をもたらす。たとえば、ＳＳＬ認証機能を備え、ウェブ・アプリケーション・サーバ、Ｌｉｎｕｘ　Ａｐａｃｈｅ、従来のアプリケーション（ＯＳ／３９０）に認証情報を提供する（ＳＡＦをＰＡＭに結びつける）ウェブ・サーバや、セキュリティ証明／コンテキストを各プラットフォーム上で公開された既存のセキュリティ・マネージャＡＰＩによって本発明の共用メモリに記憶することができるセキュリティ・マネージャ（すなわちポリシー・ディレクタまたはＲＡＣＦ）である。

【００７９】本発明の他の実施形態では、共用メモリに入れられたデータを、単一動作データ移動機構を介してカーネル・メモリ１（１６０６）とカーネル・メモリ２（１６０８）の間で移動させ、それによって共用メモリの作成を回避するだけでなく、ネットワーク接続も回避する。

【００８０】本発明のセキュリティ・サーバにおいて、第１の区分（１６１４）で共通セキュリティ・サーバ（１６０１）が稼働し、少なくとも１つの第２の区分（１６１５）で少なくとも１つのセキュリティ・クライアント（またはプロキシ）（１６０３）が稼働する区分処理システムのためにセキュリティを設ける通信ステップを実施する例を以下に示す。

【００８１】ユーザ（１６５０）が認証を要求する。ユーザは、この要求を当技術分野で周知の任意の手段によって渡す。ユーザは、たとえば、端末に接続されたキーボード、タッチ・スクリーン技法、または音声変換を使用して、要求を入力することができる。ユーザは、要求を実行の一部とするプログラムで要求を渡すこともできる。セキュリティ・クライアント（１６０３）が、ユー

ザからパスワードを受け取る。セキュリティ・クライアントは、この要求をセキュリティ・サーバ（１６１０）がアクセス可能な記憶場所に入れ、それを行ったことを通知する。第１の区分（１６１４）内の「セキュリティ・デーモン」がこの通知を認識し、第１の区分（１６１４）で「プロキシ」クライアント（１６１６）を開始する。プロキシ（１６１６）クライアントは、セキュリティ・サーバ（１６０１）固有のインタフェースを使用して、この要求でセキュリティ・サーバを呼び出す。セキュリティ・サーバ（１６０１）は、要求を処理し、プロキシ・クライアント（１６１６）にサーバ応答を返す。プロキシ・クライアントは、セキュリティ・サーバの応答を、第２の区分内のセキュリティ・クライアントがアクセス可能なメモリに入れ、それを行ったことを通知する。この通知によって、セキュリティ・クライアント（１６０３）が覚醒し、許可を指し示す。セキュリティ・クライアント（１６０３）は、この応答をユーザに返す。一実施形態では、第２の区分（１６１５）内のセキュリティ・クライアント（１６０３）は、共用メモリ・インタフェース（１６０９）を使用して第１の区分（１６１４）内のセキュリティ・サーバ（１６０１）と通信し、ネットワーク接続のセキュリティ露出を回避し、パフォーマンスを向上させる。他の実施形態では、第２の区分内のセキュリティ・クライアントは、図９に示すデータ移動機構（８２１）を使用した内部メモリ間移動によって、第１の区分内のセキュリティ・サーバと通信する。図９を参照すると、この第２の実施形態は、セキュリティ・クライアントをプロセスＡ（８０３）として実施し、セキュリティ・プロキシはプロセスＢ（８０１）として実施して、外部のネットワーク接続を回避し、共用メモリの実施も回避する。

【００８２】以上、本明細書では、好ましい実施形態を図示して詳述したが、本発明の主旨から逸脱することなく様々な修正、追加、代用などを行うことができ、したがってそれらも特許請求の範囲で規定されている本発明の範囲に入るものとみなされることが、当業者ならわかるであろう。

【００８３】まとめとして、本発明の構成に関して以下の事項を開示する。

【００８４】（１）共通セキュリティ・サーバを含む第１の区分とセキュリティ・クライアントを含む第２の区分とを有する区分処理システムにおいてセキュリティを設ける方法であって、

ａ）ユーザによる許可要求を前記第２の区分内の前記セキュリティ・クライアントに送るステップと、

ｂ）前記許可要求を前記セキュリティ・クライアントから前記第１の区分内の前記共通セキュリティ・サーバに送るステップと、

ｃ）前記許可要求に対する第１の応答を前記第１の区分内の前記共通セキュリティ・サーバから前記第２の区分内の前記セキュリティ・クライアントに送るステップとを有し、区分間の前記要求または前記第１の応答のうちのいずれか一方の前記送信が主記憶域を経由し、

ｄ）さらに、前記セキュリティ・クライアントから前記ユーザに第２の応答を送るステップとを含む方法。

（２）ステップｂの許可要求の送信が、

ｂ１）前記第２の区分内で稼働する前記セキュリティ・クライアントが信号を送って前記第１の区分で稼働している第１のプログラムに前記第１の区分内のプロキシ・クライアントを始動させるステップと、

ｂ２）前記プロキシ・クライアントから前記第１の区分内の前記セキュリティ・サーバに前記要求を送るステップとをさらに含む、上記（１）に記載の区分処理システムにおいてセキュリティを設ける方法。

（３）ステップｂまたはステップｃのいずれかが、前記第１の区分と前記第２の区分との間で共用される主記憶域を使用するステップを含む、上記（１）に記載の区分処理システムにおいてセキュリティを設ける方法。

（４）ステップｂまたはステップｃのいずれかが、メモリ間データ移動によって前記第１の区分と前記第２の区分との間にリンクされた主記憶域を使用するステップを含む、上記（１）に記載の区分処理システムにおいてセキュリティを設ける方法。

（５）ステップｂが、前記セキュリティ・サーバ固有のインタフェースを使用した前記プロキシ・クライアントによるプログラム呼出しを含む、上記（１）に記載の区分処理システムにおいてセキュリティを設ける方法。

（６）ステップｃが、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第１の区分内で稼働している第１のプログラムを介して前記第１の応答を送るステップをさらに含む、上記（１）に記載の区分処理システムにおいてセキュリティを設ける方法。

（７）前記第２の区分内の前記セキュリティ・クライアントから前記ユーザに送られる前記第２の応答がプログラム処置である、上記（１）に記載の区分処理においてセキュリティを設ける方法。

（８）共通セキュリティ・サーバを含む第１の区分とセキュリティ・クライアントを有する第２の区分とを有するセキュリティを設ける区分処理システムであって、ユーザによる許可要求を前記第２の区分内の前記セキュリティ・クライアントに送る手段と、前記許可要求を前記セキュリティ・クライアントから前記第１の区分内の前記共通セキュリティ・サーバに送る手段と、前記許可要求に対する第１の応答を前記第１の区分内の前記共通セキュリティ・サーバから前記第２の区分内の前記セキュリティ・クライアントに送る手段とを有し、区分間の前記要求または前記第１の応答のうちのいずれか一方の前記送信が主記憶域を経由し、さらに前記セキュリティ・クライアントから前記ユーザに第２の応答を送る手段とを含む区分処理システム。

（９）前記許可要求を送る前記手段が、前記第１の区分内で稼働し、プロキシ・クライアントを始動させる第１のプログラムと、前記第２の区分内で稼働している前記セキュリティ・クライアントによって前記第１のプログラムに信号を送り、それによって前記第１の区分内の前記プロキシ・クライアントを始動させる手段と、前記プロキシ・クライアントから前記第１の区分内の前記セキュリティ・サーバに前記要求を送る手段とをさらに含む、上記（８）に記載の区分処理システム。

（１０）前記主記憶域が前記第１の区分と前記第２の区分との間で共用される記憶域を含む、上記（８）に記載の区分処理システム。

（１１）前記第１の区分と前記第２の区分との間にリンクされた記憶域をさらに含み、前記セキュリティ・クライアントから前記許可要求を送る前記手段がメモリ間データ移動機構を含む、上記（８）に記載の区分処理システム。

（１２）前記プロキシ・クライアントから前記要求を送る前記手段が前記セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出しを送る手段を含む、上記（８）に記載の区分処理システム。

（１３）前記共通セキュリティ・サーバから前記許可要求に対する応答を送る前記手段が、前記第１の区分内で稼働し、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記応答を送る第１のプログラムをさらに含む、上記（８）に記載の区分処理システム。

（１４）第１の区分が共通セキュリティ・サーバを含み、第２の区分がセキュリティ・クライアントを含む区分処理システムにおいてセキュリティを設けるコンピュータ・プログラムであって、前記システムに、

ａ）ユーザによる許可要求を前記第２の区分内の前記セキュリティ・クライアントに送る手順と、

ｂ）前記許可要求を前記セキュリティ・クライアントから前記第１の区分内の前記共通セキュリティ・サーバに送る手順と、

ｃ）前記許可要求に対する第１の応答を前記第１の区分内の前記共通セキュリティ・サーバから前記第２の区分内の前記セキュリティ・クライアントに送るステップであって、区分間の前記要求または前記第１の応答のうちのいずれか一方の前記送信が主記憶域を経由する手順と、

ｄ）前記セキュリティ・クライアントから前記ユーザに第２の応答を送る手順とを実現させる、コンピュータ・プログラム。

（１５）前記許可要求を送る手順が、

ｂ１）前記第２の区分内で稼働する前記セキュリティ・クライアントが信号を送って前記第１の区分で稼働している第１のプログラムに前記第１の区分内のプロキシ・クライアントを始動させる手順と、

ｂ２）前記プロキシ・クライアントから前記第１の区分内の前記セキュリティ・サーバに前記要求を送る手順とをさらに含む、上記（１４）に記載のコンピュータ・プログラム。

（１６）前記手順ｂまたは手順ｃにおいて、前記第１の区分と少なくとも１つの前記第２の区分の１つとの間で共用される記憶域を使用する、上記（１４）に記載のコンピュータ・プログラム製品。

（１７）前記手順ｂまたは手順ｃにおいて、前記第１の区分と少なくとも１つの前記第２の区分の１つとの間にメモリ棚データ移動機構によってリンクされた記憶域を使用する、上記（１４）に記載のコンピュータ・プログラム製品。

（１８）前記手順ｂにおいて、前記セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出しを供給する、上記（１４）に記載のコンピュータ・プログラム製品。

（１９）前記手順ｃにおいて、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第１の区分内で稼働している第１のプログラムを介して前記応答を送る、上記（１４）に記載のコンピュータ・プログラム製品。

（２０）共通セキュリティ・サーバを含む第１の区分とセキュリティ・クライアントを有する第２の区分とを有するセキュリティを設ける区分処理システムであって、前記第１の区分によるアクセスが可能な第１の部分と前記第２の区分によるアクセスが可能な第２の部分とを有する主記憶域と、前記セキュリティ・クライアントに接続され、ユーザによる許可要求を前記セキュリティ・クライアントに送る機構と、前記セキュリティ・クライアントから前記共通セキュリティ・サーバに前記許可要求を送る第１の送信部と、前記第１の区分内の前記共通セキュリティ・サーバから前記第２の区分内の前記セキュリティ・クライアントに前記許可要求に対する第１の応答を送る前記共通セキュリティ・サーバ内の第２の送信部と、を有し、前記区分間の前記要求または前記第１の応答のいずれか一方の送信が主記憶域を介し、さらに、前記セキュリティ・クライアントから前記ユーザに第２の応答を送る前記共通セキュリティ・サーバ内の第３の送信部とを含む、区分処理システム。

（２１）前記第１の送信部が、プロキシ・クライアントを始動させる前記第１の区分内で稼働するプログラムと、前記プログラムに信号を送り、それによって前記第１の区分内の前記プロキシ・クライアントを始動させる前記セキュリティ・クライアントと、前記プロキシ・クライアントから前記セキュリティ・サーバに前記要求を送る第４の送信部とを含む、上記（２０）に記載の区分処理システム。

（２２）前記主記憶域が、前記第１の区分と前記第２の

区分の両方によるアクセスが可能な第３の部分を含む、上記（２０）に記載の区分処理システム。

（２３）前記第１の区分と前記第２の区分の間にリンクされた記憶域をさらに含み、前記第２の送信部が前記主記憶域の前記第１の部分と前記第２の部分との間でデータを移動するメモリ間データ移動機構を含む、上記（２０）に記載の区分処理システム。

（２４）前記第４の送信部が、セキュリティ・サーバ固有のインタフェースを使用して前記プロキシ・クライアントによるプログラム呼出を送る、上記（２１）に記載の区分処理システム。

（２５）前記第２の送信部が、前記第１の区分内で稼働し、前記共通セキュリティ・サーバから前記セキュリティ・クライアントに前記第１の応答を送るプログラムをさらに含む、上記（２０）に記載の区分処理システム。

（２６）第２のセキュリティ・クライアントを有する第３の区分をさらに含み、前記第１の区分の前記共通セキュリティ・サーバが前記第２の区分内の前記セキュリティ・クライアントまたは前記第３の区分内の前記第２のセキュリティ・クライアントからの許可要求に応答する、上記（２０）に記載の区分処理システム。

【図面の簡単な説明】

【図１】区分データ処理システムの概要を示す図である。

【図２】１つまたは複数のシステム・ボードから成る区分を有する物理区分処理システムを示す図である。

【図３】論理区分資源がそれぞれの区分に専用化された論理区分処理システムを示す図である。

【図４】論理区分資源を複数の区分間で動的に共用することができる、論理区分処理システムを示す図である。

【図５】ＵＮＩＸ（Ｒ）オペレーティング・システムの「プロセス間通信」の構造を示す図である。

【図６】スタンドアロン・ユーティリティによってロードされる構成テーブルに従って実メモリが共用される、本発明の一実施形態を示す図である。

【図７】入出力アダプタとそのドライバの機能を使用して区分間のデータ転送を容易にする、本発明の一実施形態を示す図である。

【図８】図７の実施形態の従来技術のシステムを示す図である。

【図９】区分間の実データ転送が、区分データ処理システムの通信ファブリックで実施されたデータ移動機構によって行われる、本発明の一実施形態を示す図である。

【図１０】例示のデータ移動機構の構成要素を示す図である。

【図１１】ＩＢＭ　Ｓ／３９０移動命令の例示の書式を示す図である。

【図１２】アダプタ・データ移動を行う例示のステップを示す図である。

【図１３】プロセッサ・データ移動を行う例示のステップを示す図である。

【図１４】ワークロード・マネージャ（ＷＬＭ）の高水準図である。

【図１５】典型的なワークロード・マネージャ・データを示す図である。

【図１６】間接入出力を使用したクライアント／サーバのクラスタ化を示す図である。

【図１７】クライアント／サーバのサーバ・クラスタ化を示す図である。

【符号の説明】

１００　区分処理システム
１０１　メモリ資源ブロック
１０２　プロセッサ資源ブロック
１０３　入出力資源ブロック
１０４　相互接続ファブリック
２００　物理区分処理システム
２００Ａ１　システム・ボード
２０１Ａ　メモリ
２０１Ｂ　メモリ
２０２Ａ　プロセッサ
２０２Ｂ　プロセッサ
２０３Ａ　入出力装置
２０３Ｂ　入出力装置
２０４Ａ　相互接続媒体
２０４Ｂ　相互接続媒体
２０５　相互接続ファブリック
４００　論理区分資源共用システム
４０１　メモリ
４０２　プロセッサ
４０３　入出力資源
４０６　仮想プロセッサ
４０７　入出力ドライバ
４０８　ハイパバイザ
５０１　プロセス
５０２　メモリ
５０３　プロセス
５０４　メモリ
５０５　カーネル
５０６　メモリ
５０７　メモリ
７０１　アプリケーション・プロセス
７０２　メモリ
７０３　プロセス
７０４　メモリ
７０５　カーネル１
７０６　カーネル・メモリ１
７０７　カーネル・メモリ２
７０８　メモリＫ２
７１６　デバイス・ドライバ
７１７　デバイス・ドライバ

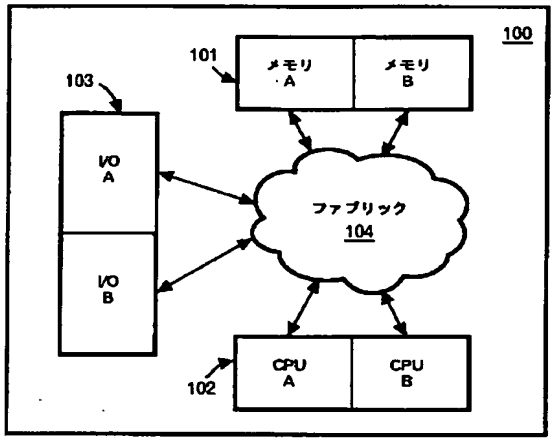
33

７１８　ソケットＡＰＩ
７１９　ソケットＡＰＩ
７２０　入出力アダプタ
８０１　プロセス
８０２　メモリ
８０３　プロセス
８０４　メモリ
８０５　カーネル
８０６　メモリ

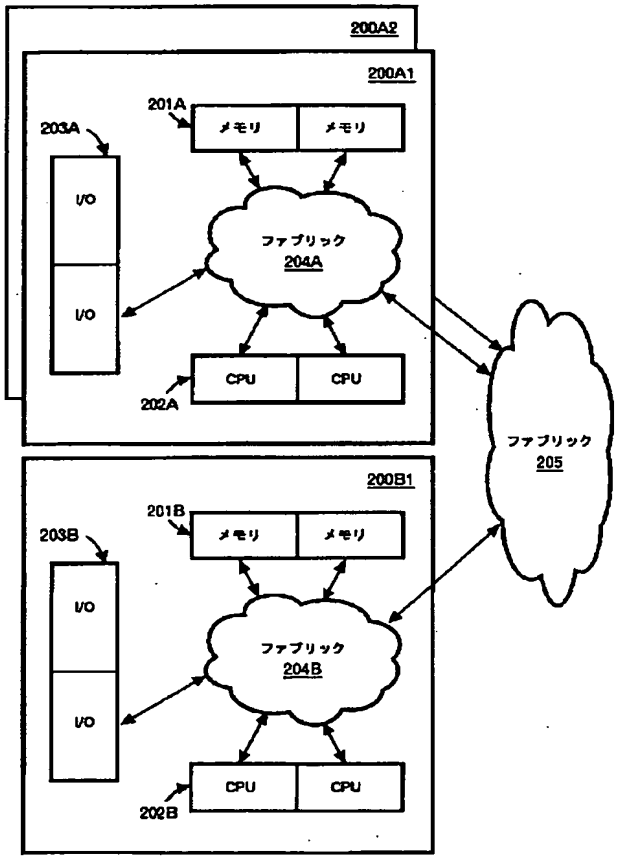
34

８０７　カーネル
８０８　メモリ
８１６　デバイス・ドライバ
８１７　デバイス・ドライバ
８１８　ソケットＡＰＩ
８１９　ソケットＡＰＩ
８２０　入出力アダプタ
８２１　データ移動機構

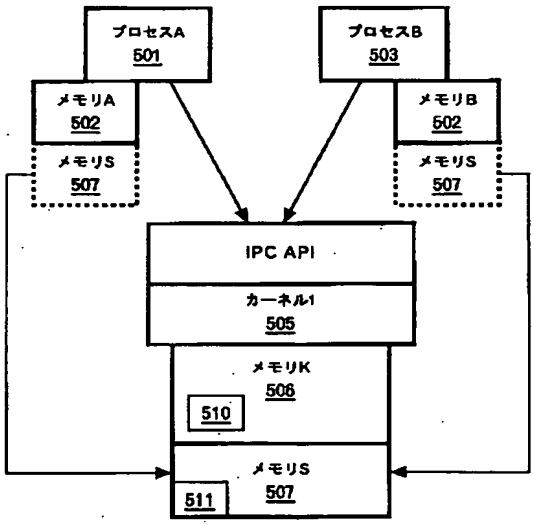
【図１】



【図２】



【図５】



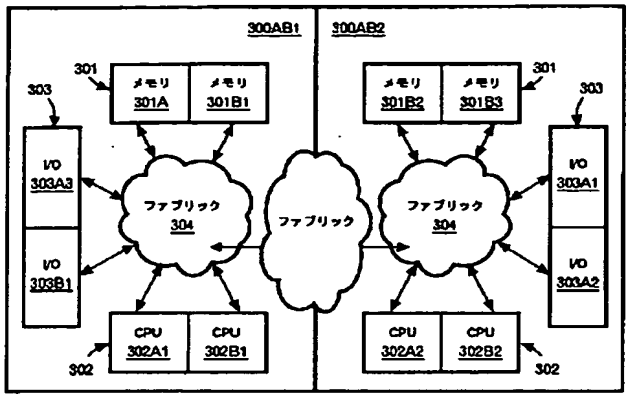
【図１１】

1000

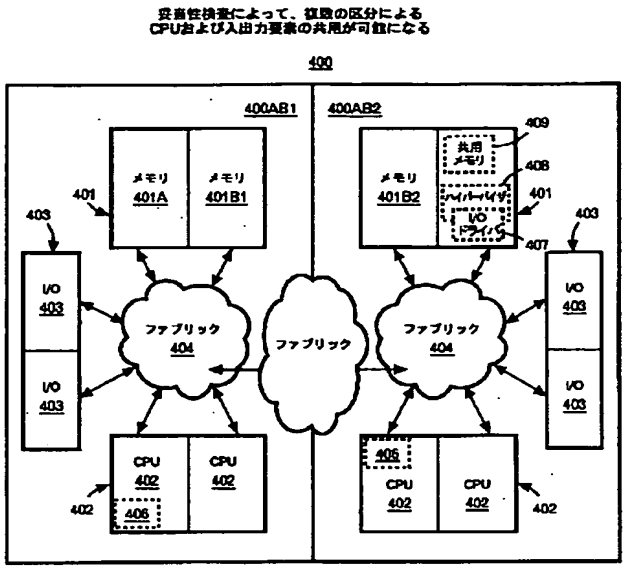
MVXL	カウント	宛先 n	ソース

MVXLが、ソース・レジスタによって指定された物理アドレスから宛先レジスタによって指定された物理アドレスに、カウント・レジスタによって指定されたバイト数を移動する
この命令は特権命令である
(MVCLが、仮想アドレス間でこれと同じ機能を実行する)
ここでは、デバイス・ドライバはレジスタに仮想ドライバではなく物理アドレスをロードし、それによって区分間データ移動を可能にする

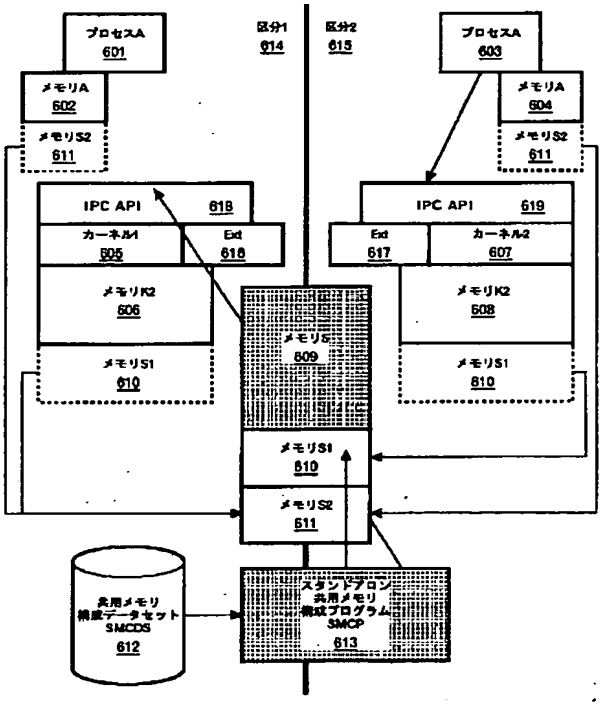
【図３】


300AB1
300AB2
301
メモリ 301A
メモリ 301B1
メモリ 301B2
メモリ 301B3
303
I/O 303A3
I/O 303B1
I/O 303A1
I/O 303A2
ファブリック 304
ファブリック
CPU 302A1
CPU 302B1
CPU 302A2
CPU 302B2
302

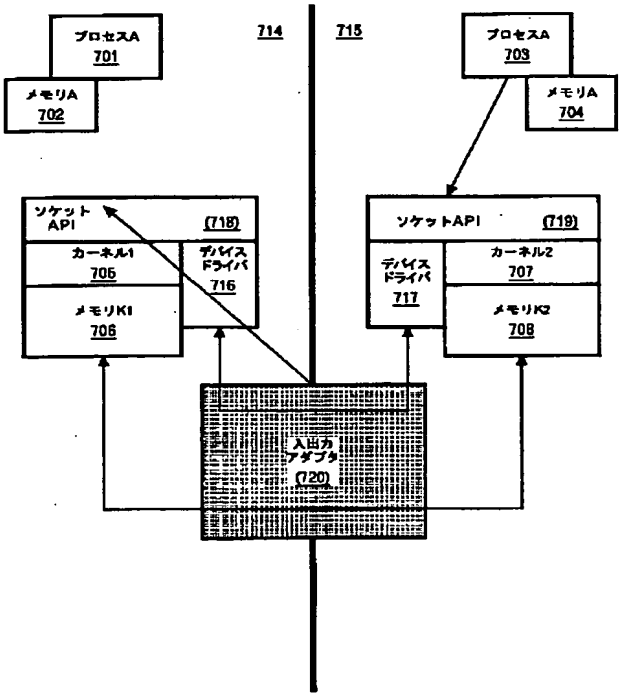
【図４】


妥当性検査によって、領域の区分による
CPUおよび入出力要素の共用が可能になる
400
400AB1
400AB2
メモリ 401A
メモリ 401B1
メモリ 401B2
共用メモリ
409
408
ハイパーバイザ
401
I/O ドライバ
407
403
I/O 403
ファブリック 404
ファブリック
CPU 402
406
402

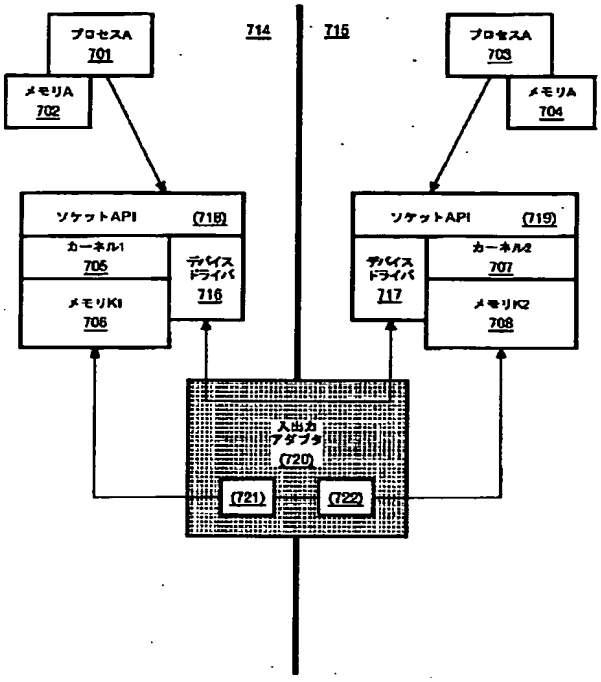
【図６】


プロセスA 601
区分1 614
区分2 615
プロセスA 603
メモリA 602
メモリA 604
メモリS2 611
IPC API 618
IPC API 619
カーネル1 605
Ext 616
Ext 617
カーネル2 607
メモリK2 606
メモリK2 608
メモリS 609
メモリS1 610
メモリS2 611
共用メモリ 構成データセット SMCDS 612
スタンドアロン 共用メモリ 構成プログラム SMCP 613

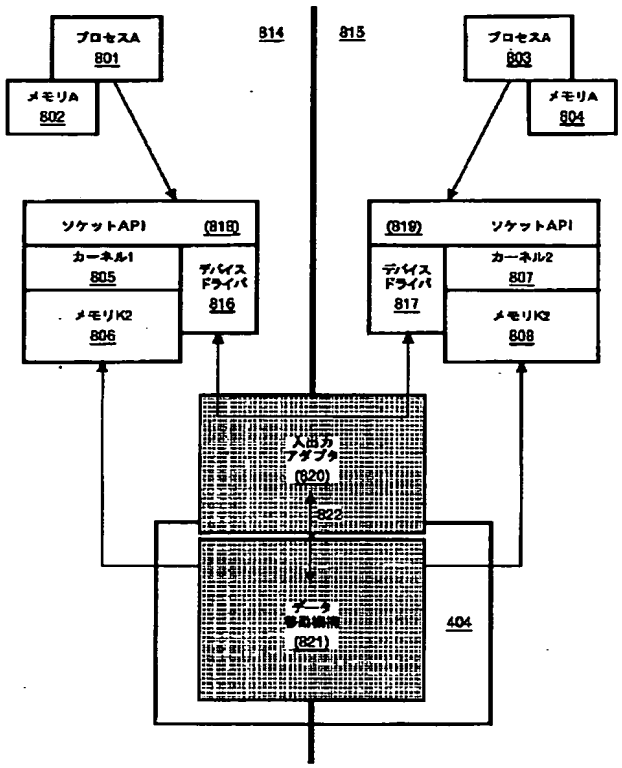
【図７】


714
715
プロセスA 701
メモリA 702
プロセスA 703
メモリA 704
ソケット API (718)
ソケットAPI (719)
カーネル1 705
デバイス ドライバ 716
メモリK1 706
デバイス ドライバ 717
カーネル2 707
メモリK2 708
入出力 アダプタ (720)

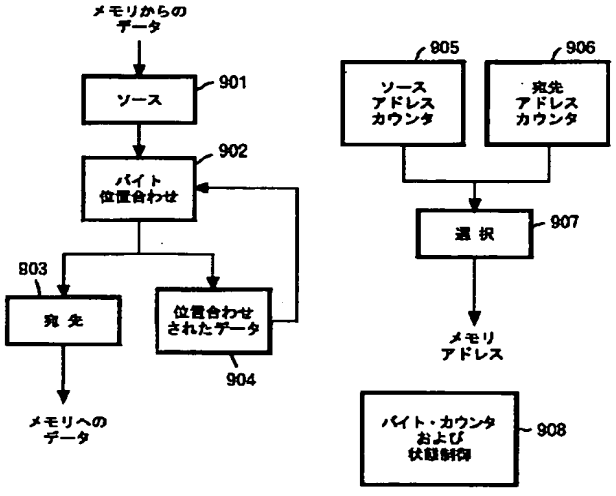
【図８】



【図９】



【図１０】



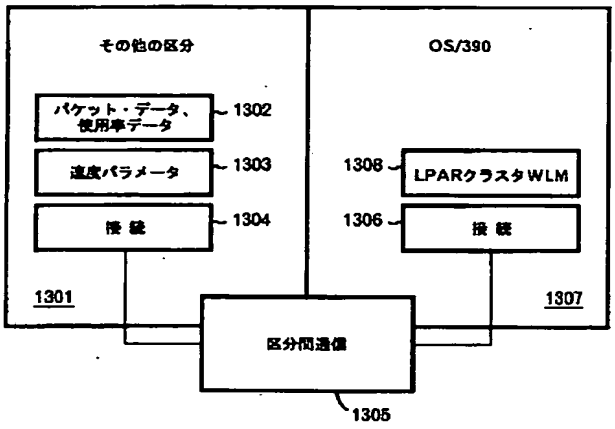
【図１２】

1101. ユーザがデバイス・ドライバを呼び出す
　－以下を供給する
　　ソース・ネットワークID
　　ソース・オフセット
　　宛先ネットワークID
1102. デバイス・ドライバがアドレスをアダプタに転送する
1103. アダプタがアドレスを変換する
　－IDから物理基底アドレスを探索する(テーブル・ルックアップ)
　－ロックおよび現行宛先オフセットを入手する
　－オフセットを加える
　－境界を調べる
1104. アダプタがカウントとアドレスをレジスタにロードする
1105. アダプタがデータ移動を実行する
1106. アダプタがロックを解放する
1107. アダプタがデバイス・ドライバに通知し、
　デバイス・ドライバがユーザに「戻る」

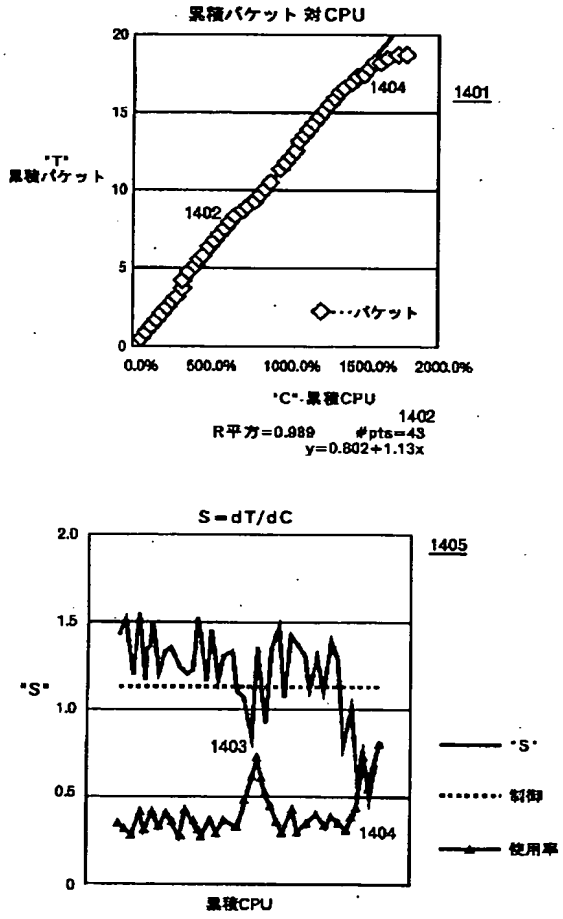
【図１３】

1201. ユーザがデバイス・ドライバを呼び出す
　－以下を供給する
　　ソース・ネットワークID
　　ソース・オフセット
　　宛先ネットワークID
1202. デバイス・ドライバがアダプタにアドレスを送る
1203. アダプタが変換する
　－IDから物理基底アドレスを探索する(テーブル・ルックアップ)
　－ロックおよび現行宛先オフセットを入手する
　－オフセットを加える
　－境界を調べる
　－ロックと物理アドレスをデバイス・ドライバに返す
1204. デバイス・ドライバがデータ移動を実行する
1205. デバイス・ドライバがロックを解放する
1206. デバイス・ドライバが戻る

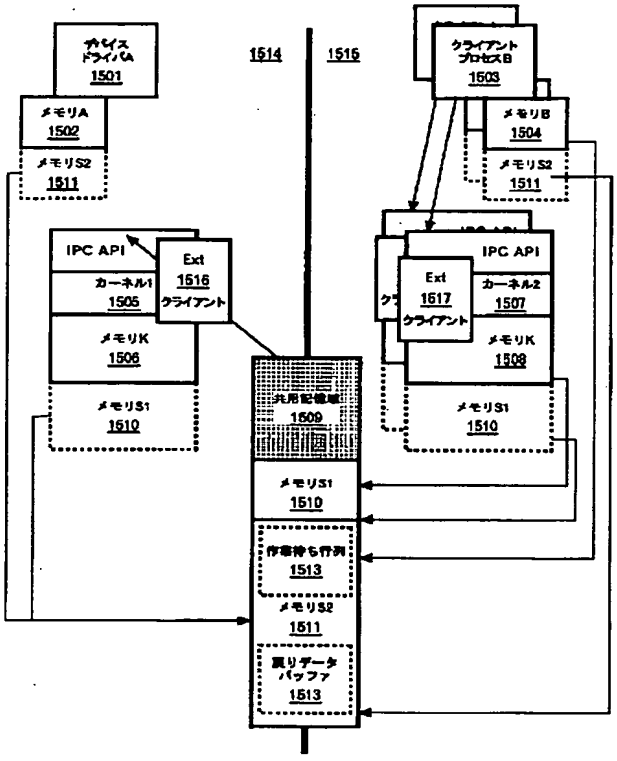
【図１４】



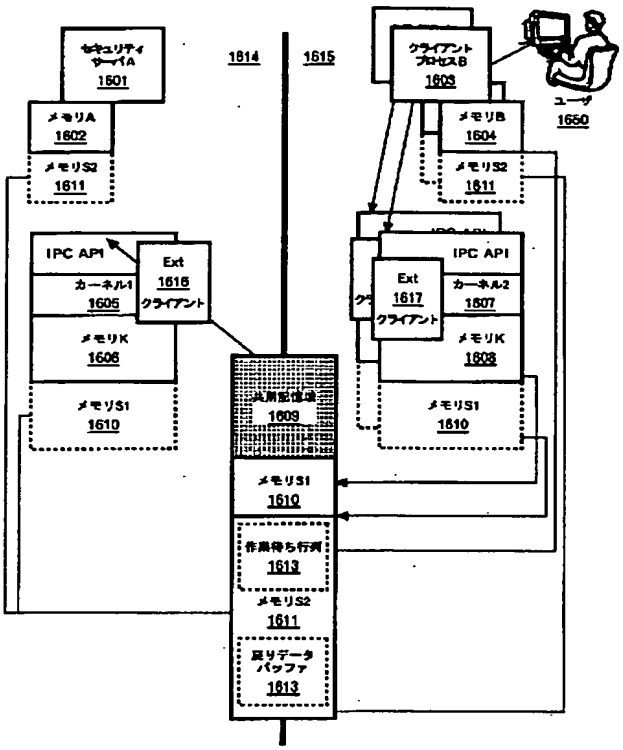
【図１５】



【図１６】



【図１７】



フロントページの続き

(72)発明者　マイケル・イー・バスキー
アメリカ合衆国12590　ニューヨーク州ワピンジャーズ・フォールズ　ハイビュー・ロード　31

(72)発明者　フランク・ジェイ・デジリオ
アメリカ合衆国12603－4621　ニューヨーク州ポーキプシー　ハイ・リッジ・ロード　13

(72)発明者　ジョン・シィ・ジョーンズ
アメリカ合衆国30062　ジョージア州マリエッタ　ジョーダン・レーク・ドライブ　4010

(72)発明者　クリスチャン・エフ・ローバッハ
アメリカ合衆国12603　ニューヨーク州ポーキプシー　オーバールック・ロード　191

(72)発明者　ジョセフ・エル・テンプル・ザ・サード
アメリカ合衆国12443　ニューヨーク州ハーリー　フック・ストリート　312　ピー・オー・ボックス　507

Ｆターム(参考)　5B045 BB28 BB42 GG01
5B085 AE00 BA06 BG07
5B089 GB01 JB14 KA17 KB13
5B098 HH04